NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

AUTOMATIC TARGET RECOGNITION (ATR)
ATR; BACKGROUND STATISTICS AND THE
DETECTION OF TARGETS IN CLUTTER

by

Nicholas Wager

December, 1994

Thesis Advisor: D. L. Fried
Co-Advisor: D. S. Davis

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE Dec. 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE AUTOMATIC TARGET RECOGNITION (ATR) ATR; BACKGROUND STATISTICS AND THE DETECTION OF TARGETS IN CLUTTER

5.

6. AUTHOR(S) Nicholas Wager

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8.

9.

10.

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*This research investigated signal processing of two dimensional signals for the detection of targets in noise, particularly in complex background pattern noise. The researchers hypothesized that this type of noise was vulnerable to non-linear processing. They investigated whether the human eye/brain acting as a surrogate for a non-linear processor could outperform an optimum linear processor in a quantitative sense. The researchers did this by conducting computer experiments to determine the ability of an operator and an optimum linear filter to determine a known pattern's presence or absence in a noisy image. The performance of both the operator and optimum linear filter are recorded as probability of detection, probability of false alarm pairs, which the researchers use to determine effective signal-to-noise ratio. The performance of man verus machine (optimum linear filter) is compared quantitatively using the effective signa-to-noise ratio. Operator and machine/filter are tested against circular targets in Random White Gaussian noise and in sattellite images. The researchers report that the machine /filter outperforms the man when the details of both target and background are known in advance, but the man outperforms the machine/filter when the details are known only in a statistical sense.

14. SUBJECT TERMS Automatic Target Recognition (ATR), probability of detection probability of false alarm pair, signal-to-noise ratio effective , non-linear filterring	15. NUMBER OF PAGES 154
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

AUTOMATIC TARGET RECOGNITION (ATR)
ATR; BACKGROUND STATISTICS AND THE
DETECTION OF TARGETS IN CLUTTER

by

Nicholas Wager
Major United States Army
B.A., New York University, 1981

Submitted in partial fulfillment
of the requirements for the degree of

Accesion For	
NTIS CRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

MASTER OF SCIENCE IN APPLIED PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
December 1994

Author: Nicholas Wager
Nicholas Wager

Approved by: David L Fried
D. L. Fried, Thesis Advisor

David Scott Davis
D. S. Davis, Co-Advisor

W. B. Colson
W. B. Colson, Chairman
Department of Physics

ABSTRACT

This research investigated signal processing of two dimensional signals for the detection of targets in noise, particularly in complex background pattern noise. The researchers hypothesized that this type of noise was vulnerable to non-linear processing. They investigated whether the human eye/brain acting as a surrogate for a non-linear processor could outperform an optimum linear processor in a quantitative sense. The researchers did this by conducting computer experiments to determine the ability of an operator and an optimum linear filter to determine a known pattern's presence or absence in a noisy image. The performance of both the operator and optimum linear filter are recorded as probability of detection, probability of false alarm pairs, which the researchers use to determine effective signal-to-noise ratio. The performance of man verus machine (optimum linear filter) is compared quantitatively using the effective signal-to-noise ratio. Operator and machine/filter are tested against circular targets in Random White Gaussian noise and in sattellite images. The researchers report that the machine /filter outperforms the man when the details of both target and background are known in advance, but the man outperforms the machine/filter when the details are known only in a statistical sense.

TABLE OF CONTENTS

ACKNOWLEDGMENT

The author would like to thank Prof. David L. Fried for his untiring, patient, and competent leadership. He has served not only as an advisor but more importantly as a teacher, counselor, mentor, and role model. The author would also like to thank Prof. Davis, Colin Cooper, and commander Petho for their assistance.

I. INTRODUCTION

It would be a major advance in the state of the art to find an algorithm for Automatic Target Recognition (ATR) which could consistently perform at the same level as a trained human observer who is given the opportunity to leisurely examine a small field of view.

Prof. David L Fried.

Automatic Target Recognition (ATR) for visible/IR images is a subject of interest to the military. A lot of work has been done. A problem exists in that this work progresses in an ad hoc manner without a clear theory. Today we do not have the ability to produce ATR hardware for the field. We believe this thesis will help provide some of the theoretical basis for future research.

In Appendix A of this thesis we review some of the work currently being conducted in the ATR community. In reviewing the work already done in ATR we noted that it almost completely ignores the types of filters which theory does predict should, in a certain sense, be optimum. To discuss these filters we reference some basic signal processing theory. In signal processing theory when considering Gaussian white noise the matched filter is the optimum filter. When considering Gaussian non-white noise the optimum filter is the matched filter Fourier transformed divided by the Power Spectrum Density (PSD), and then inverse Fourier transformed. Whether the noise is white or non-white, so long as it is Gaussian, the optimum filter is a linear filter. In reviewing the ATR work we have noted that it has almost entirely ignored optimum linear filters.

We began work with the hypothesis that the nature of background clutter is such that a machine equipped with an optimum non-linear filter would recognize targets in a noisy/cluttered background better than a machine equipped with an optimum linear filter. We hypothesized that a close approximation to the optimum non-linear processor searching for a known type of target in a noisy/cluttered background is provided by the human eye/brain, when the eye/brain is allowed to operate in a leisurely manner. We assume that after examining a reasonable number of training samples the eye/brain combination should be considered to be able to achieve near optimum performance.* If we accept the human as a near optimum non-linear filter this permits us to compare

* The near optimality of the eye/brain combination is a hypothesis which we will not attempt to prove. It certainly seems plausible, and is the underlying hypothesis in current work attempting to develop a neural net machine for ATR. In any case the ATR-customer community would certainly be pleased to develop an ATR able to perform as well as a human at leisure. The inability of curent ATR systems to match a human's performance is the reason that Dr. Rudolph G. Buser and Alexander Akerman III have said "if an ATR doesn't work near perfectly then the human operators (referring to pilots and image analysts) will just turn the system off." Accordingly we feel quite comfortable in taking the performance capability of the human eye/brain combination working at

performance of linear processors (which we know how to implement on a computer) with a near optimum non-linear filter (the human eye/brain combination).

We had expected the human, because of his ability to use non-linear processing, to outperform a machine using an optimum linear filter. We undertook to prove that this was the case since we could not find a proof in the literature. We have since come to a somewhat different conclusion.

We conclude that the optimum linear filter matches or outperforms the human surrogate non-linear filter in both the white and non-white backgrounds when the details of the target are known. We also conclude that the human outperforms the machine/filter when the target details are known only in a statistical sense. In carrying out this work we came to the following conclusions:

Conclusions

1. When the background noise was Gaussian and white the linear filtering process outperformed the human, but only by about 0.6 dB.
2. When the background noise corresponded to satellite (SPOT) imagery of the ground, for a well camouflaged target* a human's performance was inferior (or for small targets, at best, equivalent) to that of a machine that used an optimum linear filter.†
3. We observed that a human was able to detect background pattern anomalies of a type which we would associate with imperfect camouflage. The machine equipped with the optimum linear filter was unable to take note of these anomolies.‡

Conclusion 1 presented no major surprise. The signal processing concept is presumably the same for both man and machine. The performance of the man falls short, but only by about 0.6 dB. We were pleasantly surprised by the results referred to in Conclusion 2. Although it contradicted our original hypothesis, it is seen as a positive result. It tells us what filter to use for optimum (time unconstrained) performance when the target is of a known form. This thesis suggests that researchers should place more emphasis than is apparently being given to the investigation of optimum linear

---

a leisurely pace as our reference point.

* We consider a well camouflaged target to be one which does not "break" the natural pattern of the background clutter but only adds its additional signature component to the image scene.

† We understand that exploitation of conclusion 2 introduces the "curse of dimensionality," that there are very many target image patterns to be looked for. However, with the regular and exponential growth of the amount of computing power that has been seen for the past decade and we feel can be projected for the next decade, it seems to us that emphasis in the ATR development problem has to be put on developing an algorithm that works well enough. Only then should the major part of the effort be put into getting the algorithm to run fast enough.

‡ Conclusion 3 noted above supports our original hypothesis. We feel that exploitation of a non-linear process that can imitate a human provides a potentially highly effective approach to the ATR problem.

filters. Conclusion 3 indicates that non-linear processing may provide better ATR performance than linear processing under the right circumstances. We recommend that researchers further examine the human non-linear process in follow-on work.

We designed experiments to test our hypothesis using the following methodology:

1. We presume that the eye/brain combination can be as non-linear as optimality requires and so can use a human as our surrogate for the optimum non-linear filter. The level of ongoing work with (ATR) neural nets and a general consensus in the research literature leads to the conclusion that the human eye/brain combination's performance can be considered to be acceptably close to optimum (if allowed adequate time).
2. We know how to write algorithms for an optimum linear filter and so can put it on a machine such as a computer.

These two items of methodology allow us to conduct a comparison of the linear and non-linear optimum filters. Our methodology consists of testing humans (to whom we give lots of time to make a decision) and machines against the same class of target/background images. We then compare their performance to decide the merit of a non-linear filter.

The basic experiment consists of two parts. The first part involves showing a human target/background patterns displayed on a computer monitor and recording his decisions (i.e., the human decides if the target is present or if the target is not present). We (or more accurately stated the computer operating as the control center of the experiment) uses the operator's inputs to determine his Probability of Detection (P_{D}) and his Probability of False Alarm (P_{FA}). The P_{D} and P_{FA} considered together form what we call a (P_{D},P_{FA})-pair. We (i.e., the computer) then conduct a similar experiment with the machine/filter using image data sets with the same kinds of background and target definition/parameters that were used to generate the images shown to the human. These data sets are presented to a separate program in the computer which, based on an optimum linear filter and some threshold setting, implements what we call the machine/filter. The machine/filter decides if the target is present or not present. From the machine/filter's decisions we determine the machine/filter (P_{D},P_{FA})-pair — doing this for various threshold settings. For each threshold setting we compute a separate (P_{D},P_{FA})-pair. When plotted this set of pairs forms a curve called the Receiver Operating Characteristic (ROC).

It is useful to present (P_{D},P_{FA})-pairs and the ROC on a specialized graph which we have called a probability-probability (PRBPRB) graph. Since the PRBPRB graph is not commonly used we

will discuss it in enough detail for the reader to understand the data presented in this thesis.

The PRBPRB graph shows probability of detection on the vertical axis and probability of false alarm on the horizontal axis. These axes are not linearly spaced. They are spaced in such a way that any change in probability corresponding to one standard deviation (of a Gaussian probability distribution) always represents the same axis length. An example of a PRBPRB graph without any data plotted is shown in Fig.1. We use these PRBPRB plots through out the thesis to present, analyze and compare experimental results.

We next discuss the probability concept and some commonly used graphical representations which the reader will probably recognize. Fig.2 is a plot on a linear scale of the probability densities inferred from a set of samples drawn from a normal distribution. It is the familiar bell shaped curve centered at zero. The reason we don't use it to plot data is that it is difficult to form a comparison of experimental results from it. For example, we can not use it to approximate P_{D} or P_{FA}. We want a better way to graphically present experimental data, which is in the form of (P_{D},P_{FA})-pairs.

Fig.3 shows a plot of cumulative probability densities plotted on a linear scale. The cumulative probability densities are generated from the same set of Gaussian normal distribution sample values used for Fig.2. The cumulative probability distribution plot has linear vertical and horizontal axes. It is also difficult to use these graphs to display (P_{D},P_{FA})-pairs nor is it clear how one might use such a graph to form an estimate of the Effective Signal-To-Noise Ratio (SNR_{eff}) that can be associated with the (P_{D},P_{FA})-pair.

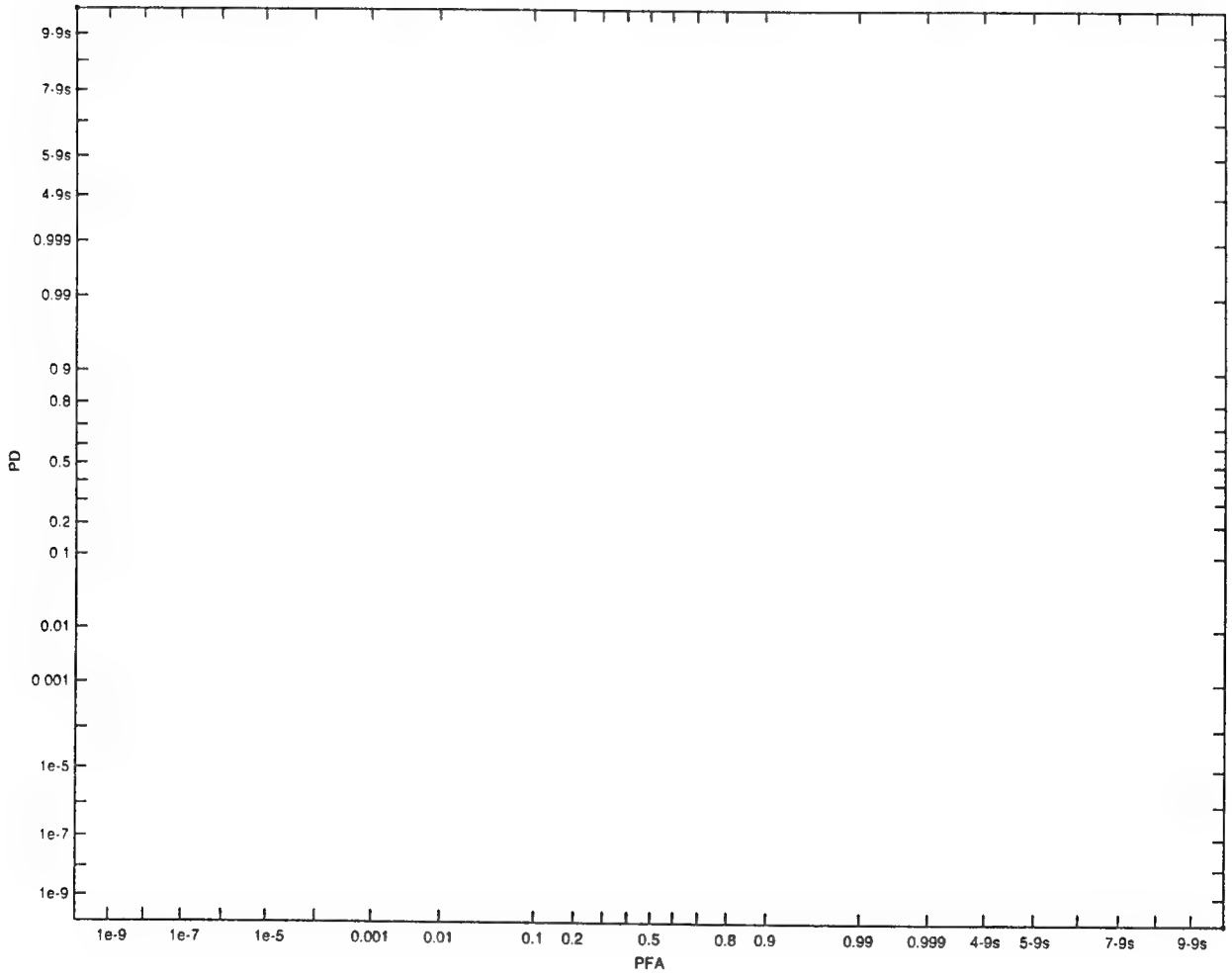


FIGURE 1. PRBPRB Graph. The above figure is a PRBPRB graph without any data plotted. The reader may not be familiar with the axis convention. The axes are spaced evenly by standard deviations of a Gaussian about a mean value of 0.5 at the midpoint corresponding to zero standard deviations.

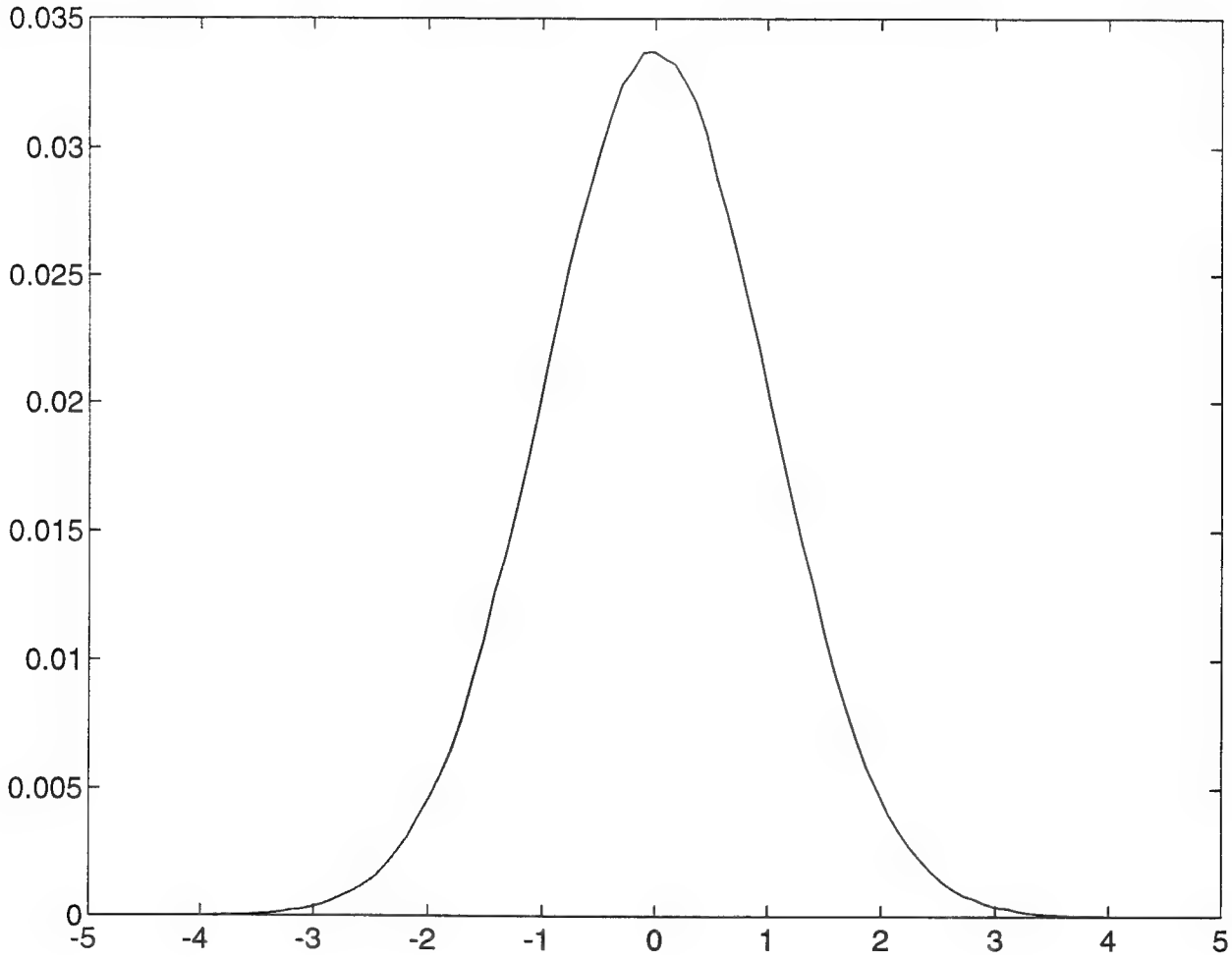


FIGURE 2. Bell Shaped Curve. The above figure shows a linear plot of the probability densities of a random normal distribution. It is the familiar Gaussian curve. The results shown are based on a set of 90,000 randomly chosen values.

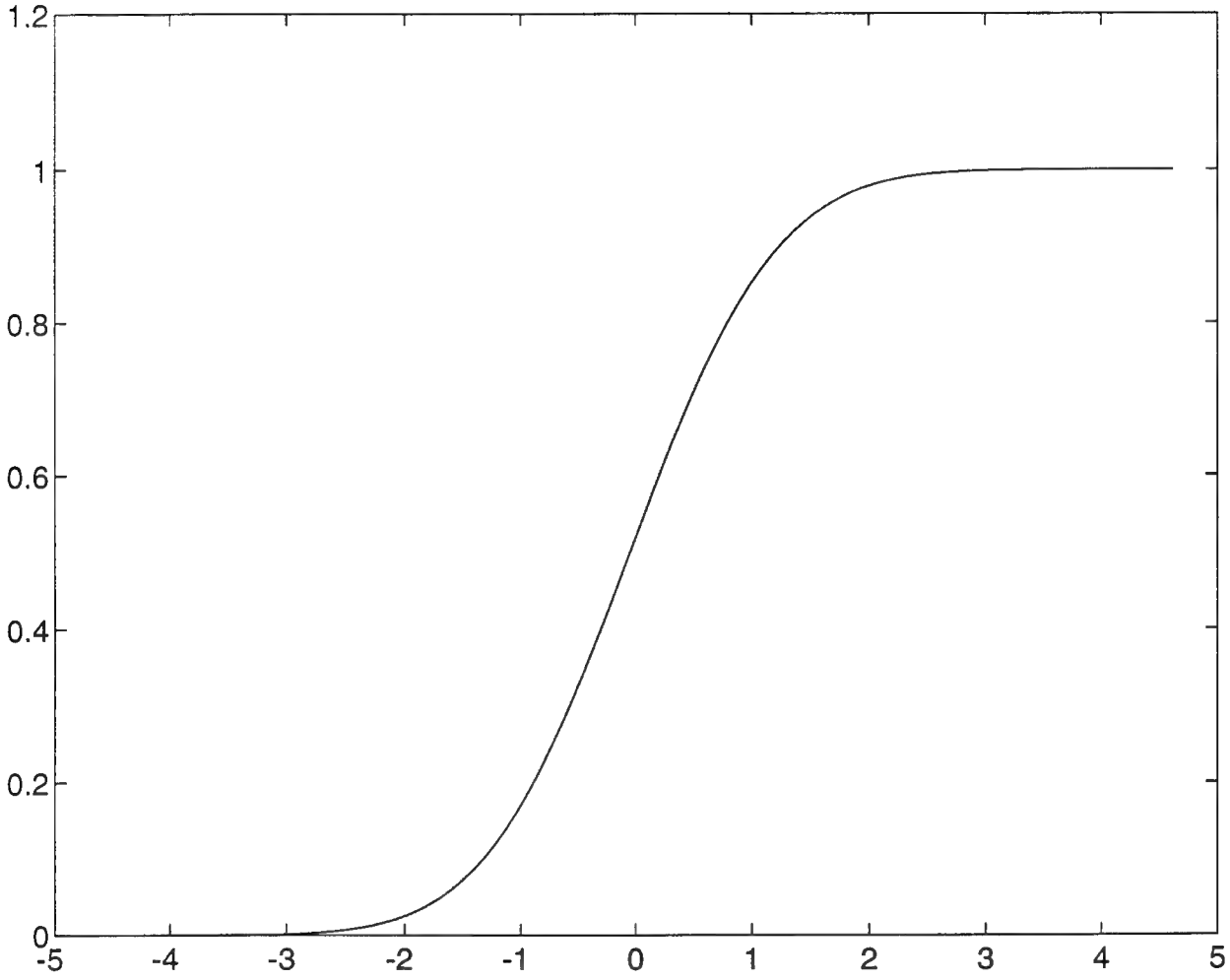


FIGURE 3. Cumulative Probability Distribution. This figure shows a plot of a cumulative probability distribution plotted on a linear scale. The plot has a distinctive S shape. Note that probabilities of interest for ATR (values near zero and one) are squeezed together somewhere on the nearly horizontal portions of the curve.

For example, try to compare a probability (vertical axis) of 0.99 with a probability of 0.999. Examine the plot and you will note that these probabilities appear squeezed together on the nearly flat horizontal portion of the graph. The difference in performance of an ATR which achieves a (P_D) of 0.999 and one which achieves a P_D of 0.99 [given that they both achieve the same (P_{FA})] is very significant. This inability to adequately represent the significant difference in probabilites that are of interest to those researching the ATR problem on a linear graph drove us to use a more representative graph.

Fig.4 shows the same cumulative probability densities of Fig.3 plotted on a graph which uses a linear scale for the horizontal axis and a probability scale for the vertical axis (called a LINPRB plot). It is easy to see on this graph the difference between probability spacing (vertical) axis and linear spacing (horizontal) axis. For instance,examine the point 0.5 on the horizontal axis and the vertical axis. On the vertical axis (which is probability spaced) the distances between tick marks are evenly spaced by their percent standard deviation from the Gaussian distribution's mean value (in this case 0.5). The effect of plotting the cumulative probability densities of a normal distribution on this graph is that they form a nearly straight line as can be seen in Fig.4. The plot also creates more "working space" for the interesting probability values near zero and one (i.e., probabilities that correspond to desirable P_{FA} and P_D respectively). Look at the graph. There is a lot of space between a probability with a value of 0.99 and another probability with a value of 0.999. The same effect is true for probabilities near zero, which we would be interested in if we wished to recognize, for example, the difference between $P_{FA}=1.0 \times 10^{-3}$ and $P_{FA}=1.0 \times 10^{-}5$. The use of such a probability scale provides the most useful convention for plotting probability data resulting from our experiments.

The next set of figures illustrates how we compare probability data using PRBPRB graphs. For a PRBPRB graph both the horizontal and the vertical axes have probability scales. We mentioned earlier that each experiment yielded one operator (P_D,P_{FA})-pair and a set of machine (P_D,P_{FA})-pairs, constituting the ROC, as data. We next discuss the presentation of this data.

In Fig.5 a typical operator (P_D,P_{FA})-pair is plotted. The reader should be able to read the operator (P_D,P_{FA})-pair from the graph. From Fig.5 we approximate P_D as 0.92 and the P_{FA} as 0.08. The reader will note that use of the probability scales provides better resolution of (P_D,P_{FA})-pairs than a linear scale. The scale is essentially (usefully) expanded at the places of interest. The use of the probability scales provides a convenient way of associating an SNR_{eff} with a (P_D,P_{FA})-pair.

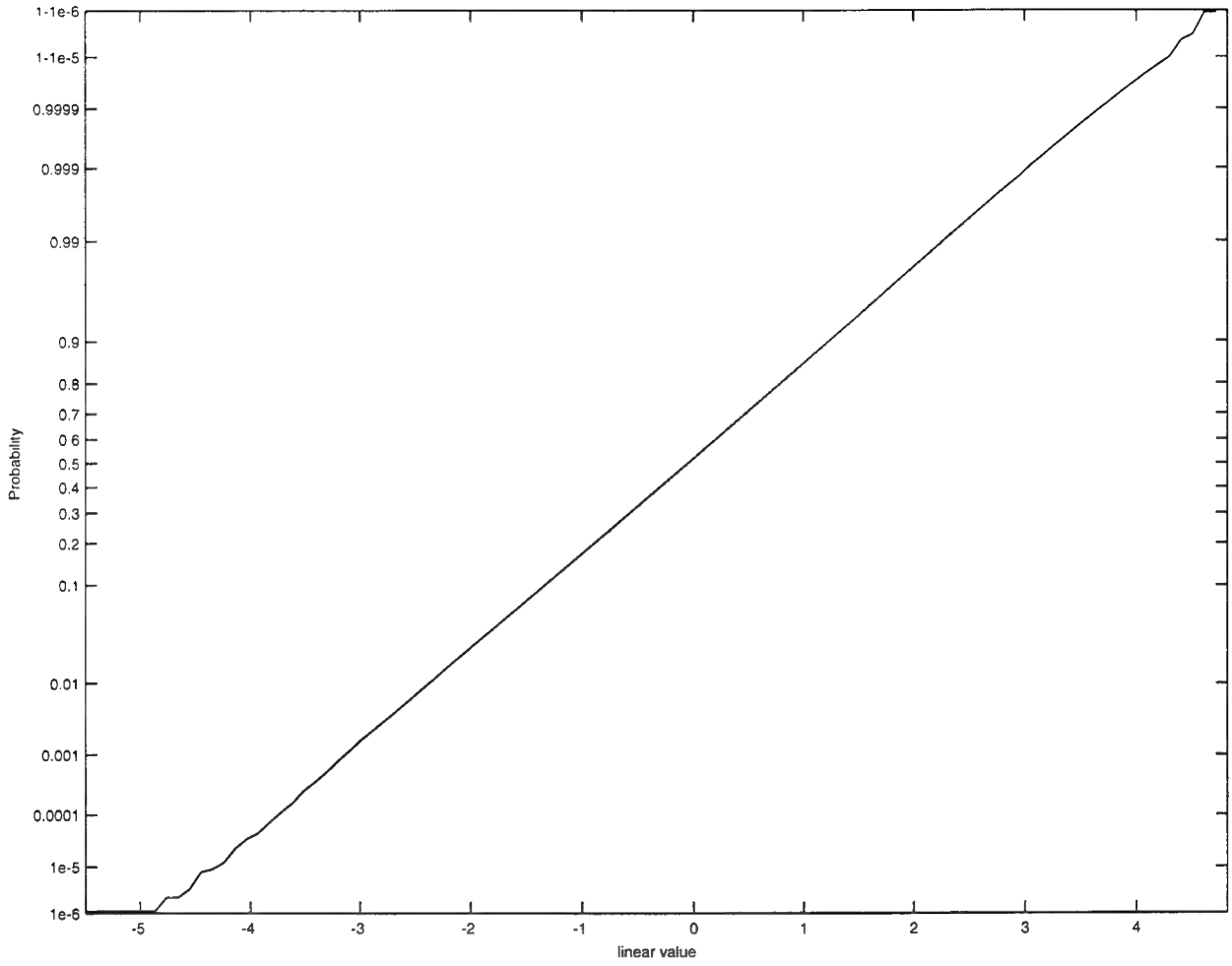


FIGURE 4. Cumulative Probability Plot. The same data is plotted here as is plotted in Fig.3. The above figure shows the cumulative probability distribution plotted on a probability scale. This has the effect of straightening the S shaped curve appearing on the previous linear plot. Note the curve is a straight line with working space at the points of interest, near zero and one.

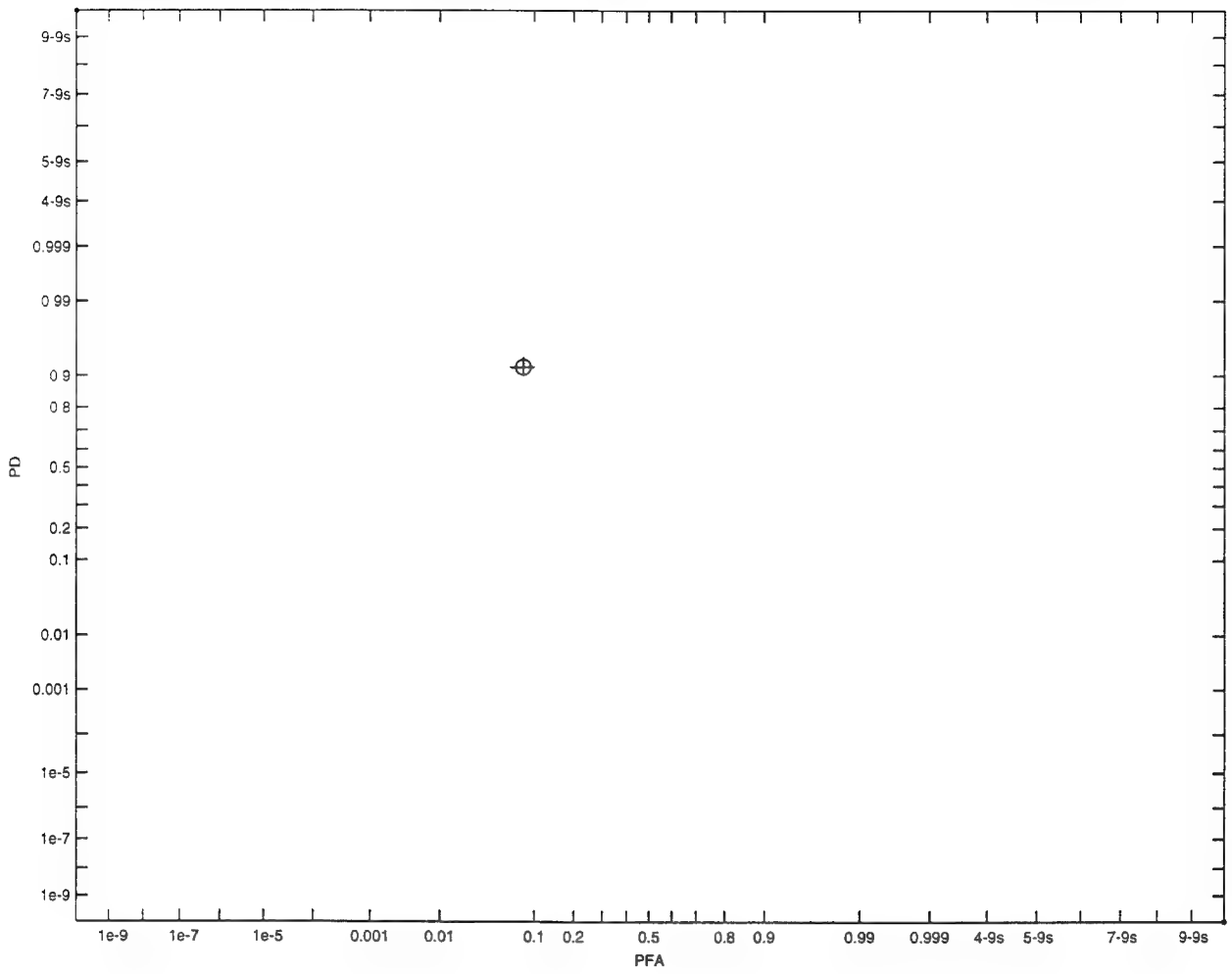


FIGURE 5. PRBPRB Plot Of Operator (Probability Of Detection, Probability Of False Alarm)-Pair. The reader should convince himself that the operator probability of detection read off the graph is 0.92 and the operator probability of false alarm is 0.08. This plot also shows the uncertainty in the operator point with error lines. The operator(probability of detection, probability of false alarm)-pair is marked with an o. The error bars, corresponding to plus and minus one standard deviation, are the horizontal and vertical lines through the o.

In Fig.6 we plot machine performance on the PRBPRB graph. In the experiments machine data was collected as a set of (P_D,P_{FA})-pairs, each pair corresponding to a different threshold (i.e., corresponding to a different balance between missed detections and avoiding false alarms). We plotted these (P_D,P_{FA})-pairs yielding the machine ROC. The ROC is shown as a short curved line in Fig.6.

In Fig.7 we show the results of a hypothetical experiment. Data is shown for a machine/filter and for an operator. The results for machine/filter and for the operator are plotted using a PRBPRB graph. Without discussing the results of the experiments at this time we point out that better performance results in data plotted up and/or to the left corresponding to higher P_D and lower P_{FA}. Weaker performance gets plotted lower and/or to the right — corresponding to lower P_D and higher P_{FA}. For Fig.7 we would conclude that the machine/filter outperforms the operator.

You might be interested to see the same data plotted on a linear graph. Fig.8 shows this. Notice that the points of interest are squeezed together at zero and one, making it difficult to read (P_D,P_{FA})-pairs.

When we consider the comparison of machine performance with operator performance we recognize that the machine performs better than the operator. We then ask the question "how much better?" We can quantitatively answer this question by adopting the technique of associating an (SNR_{eff}) with a (P_D, P_{FA})-pair. This SNR_{eff} provides the basis for comparing machine and operator performance and assessing how much better the performance of one is relative to that of the other.

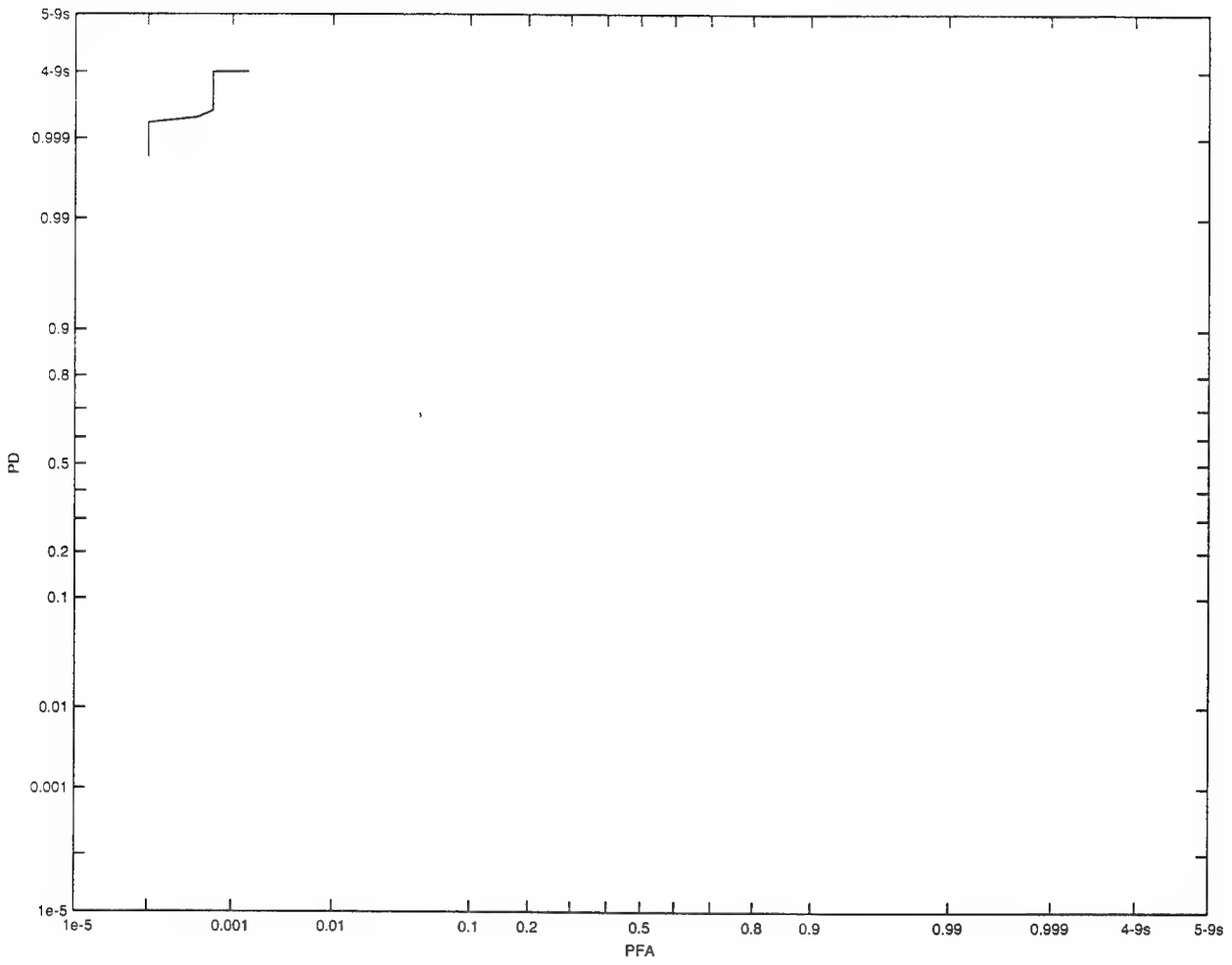


FIGURE 6. PRBPRB Graph Of Machine/Filter (Probability Of Detection, Probability Of False Alarm)-Pairs. The above graph shows machine/filter (probability of detection, probability of false alarm) pairs displayed on a PRBPRB plot.

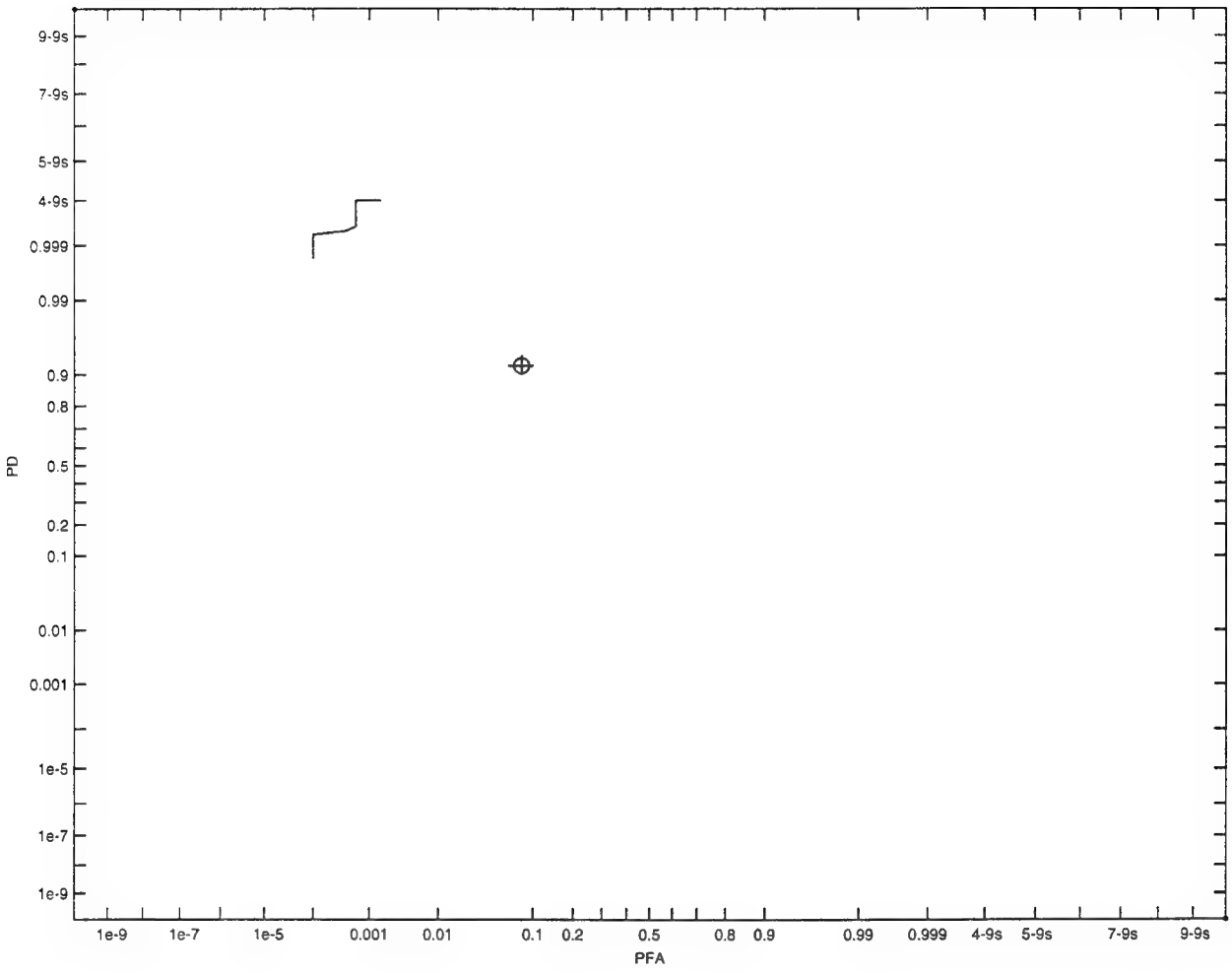


FIGURE 7. PRBPRB Graph Of Machine/Filter (Probability Of Detection, Probability Of False Alarm)-Pairs. The above graph shows machine/filter (probability of detection, probability of false alarm) pairs displayed on a PRBPRB plot.

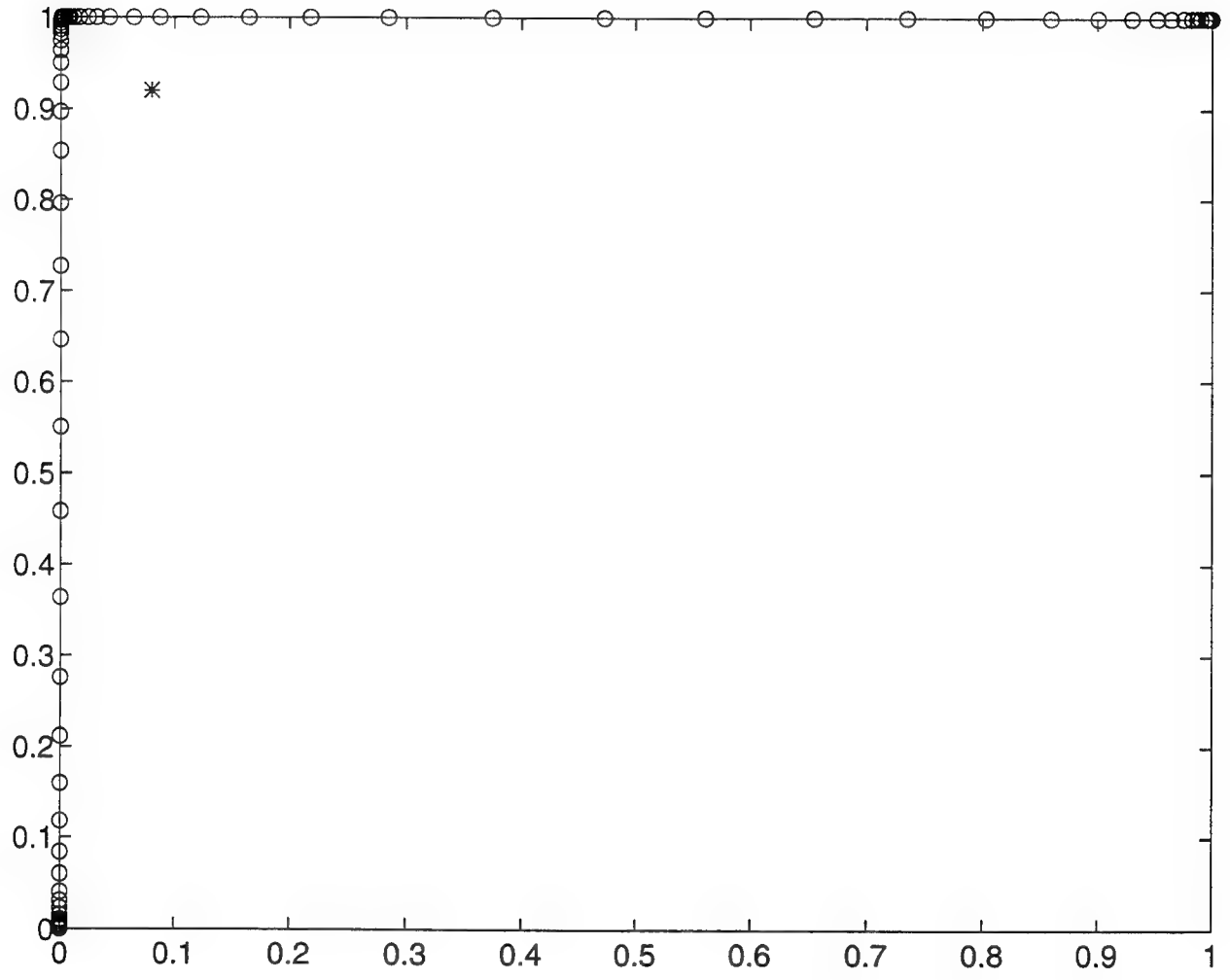


FIGURE 8. Operator, Machine/Filter Linear Plot. Note that the points of interest are clumped about zero and one and are difficult to read. The same data is shown as is plotted in Fig. 7.

The significance of the variable SNR_{eff} can best be understood by considering a signal along with Gaussian noise with various Signal-To-Noise Ratios (SNR's). For a given SNR we get a set of (P_{D},P_{FA})-pairs (each pair corresponding to some particular threshold setting) and from such a set of (P_{D},P_{FA})-pairs we form a curve. Such a curve of (P_{D},P_{FA}) will be a straight line when plotted on a PRBPRB graph as a consequence of the noise having a Gaussian distribution. Fig. 9 shows ten such (P_{D},P_{FA}) curves, each curve corresponding to a different target signal strength and thus a different SNR. The line farthest down and to the right represents the smallest SNR_{eff} (SNR_{eff}=1). The line farthest up and to the left represents the highest SNR_{eff} (SNR_{eff}=10). When we examine these Gaussian Signal-To-Noise Ratio lines (which we shall call "GLINES") we see that we can associate a (P_{D},P_{FA}) pair with a particular SNR_{eff}. We associate a higher SNR_{eff} with a better receiver operating characteristic and better performance.

We shall use these curves to estimate SNR_{eff} remembering that we must have a (P_{D},P_{FA})-pair to estimate SNR_{eff}. In fact, given any two of the three quantities SNR_{eff}, P_{D}, and P_{FA} we can use the PRBPRB graph to estimate the third. We can read the remaining quantity from the graph.

We use these PRBPRB graphs with the GLINES to plot (P_{D},P_{FA}) data and infer SNR_{eff} values from our experiments. We then compare the SNR_{eff} of the operator with the SNR_{eff} of the machine. For instance, in Fig.10 we have plotted the set of machine (P_{D},P_{FA})-pairs (the curved line) and the operator (P_{D},P_{FA})-pair (the circle) on a PRBPRB graph with GLINES. We can approximate from the GLINES that the operator SNR_{eff} is about 2.8 and that the machine SNR_{eff} is about 7.0 (reading the max value). Since these are signal-to-noise voltage ratios the difference corresponds to $20\log_{10}$ (7/2.8) or about 8.0dB. This example shows how the PRBPRB graph with the GLINES can be used to approximate the SNR_{eff} for both the operator and the machine. We use the SNR_{eff} as a measure of performance and the dB difference between two SNR_{eff} values to compare performance — higher SNR_{eff} being associated with better performance. We think this is an unambiguous and useful technique and recommend that the ATR community consider adopting it or some similarly unambiguous technique as a convention for discussing and comparing ATR machine/filter performance.

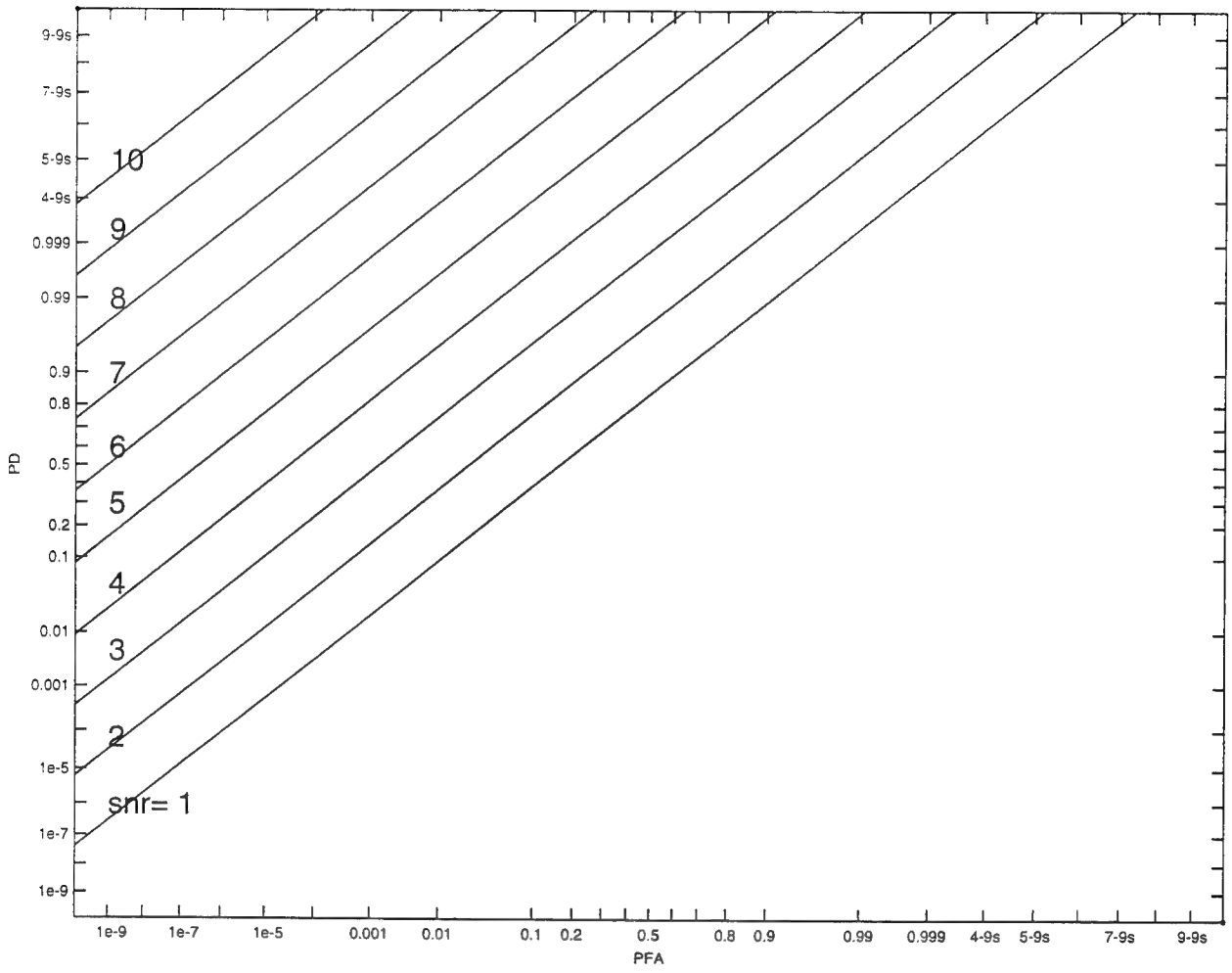


FIGURE 9. PRBPRB Graph With Gaussian Equivalent SNR Lines. The above figure depicts ten lines corresponding to signal-to-noise ratios from one (labelled in lower left corner) to ten (labelled in upper left). The horizontal and vertical axes use probability scales.

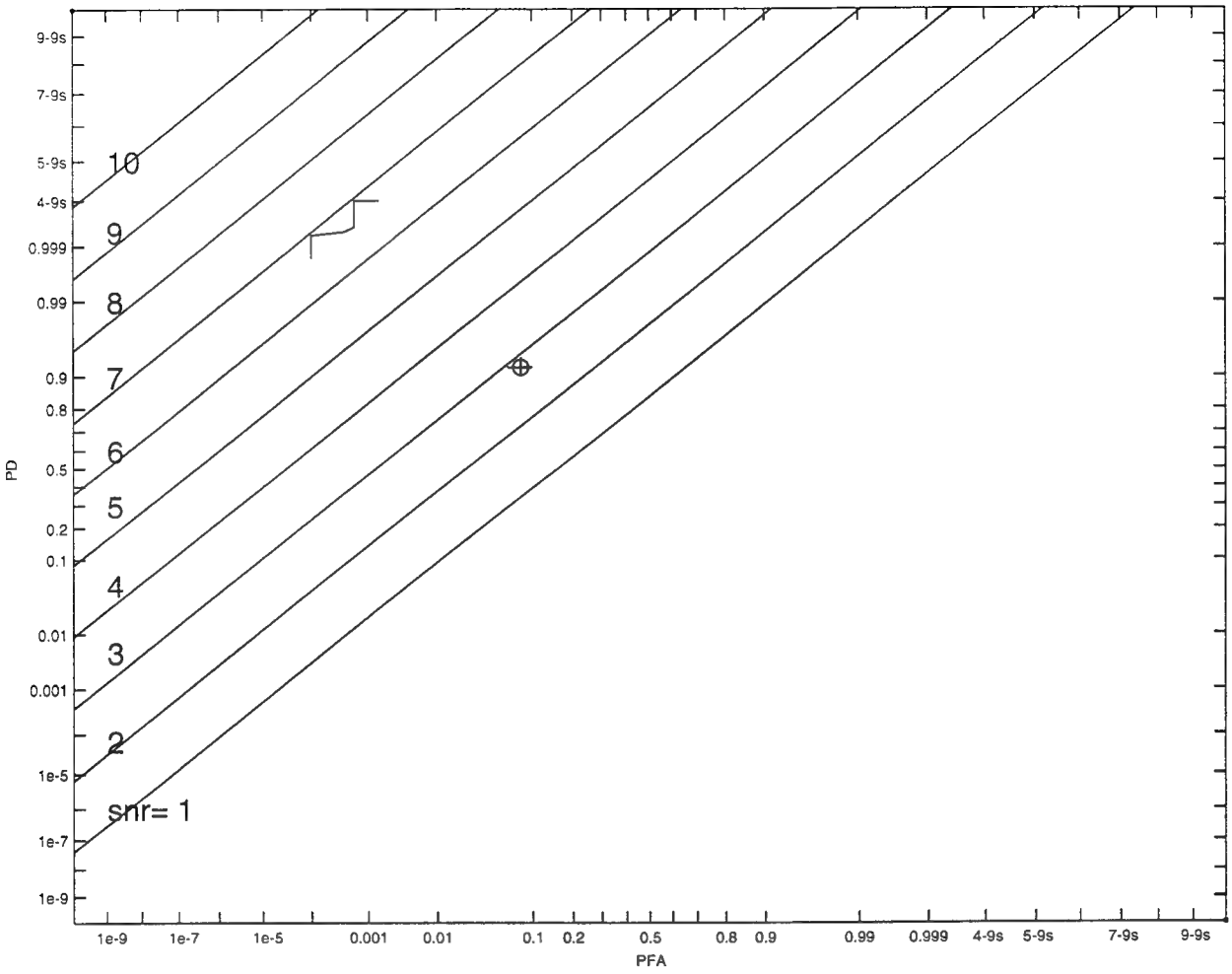


FIGURE 10. PRBPRB Plot With Gaussian Equivalent SNR Lines Of Operator And Machine Performance. The above graph shows an operator (probability of detection, probability of false alarm)-pair and a set of machine (probability of detection, probability of false alarm)-pairs on a PRBPRB graph with GLINES. The GLINES are labelled from the lowest(SNR=1) to the highest(SNR=10). In the above example the machine has an effective signal-to-noise ratio of about 7.0 and the operator has an effective signal-to-noise ratio of about 2.8. Thus the operator performance is about 8dB down from that of the machine/filter. The PRBPRB graph with GLINES is useful for approximating effective signal-to-noise ratio, which we use as a measure of performance.

The SNR_{eff} can also be calculated analytically. The formulaes used to analytically calculate SNR_{eff} are discussed next. This analytic calculation of SNR_{eff} is more accurate than the graphical approximation. SNR_{eff} is calculated analytically from a (P_{D}, P_{FA})-pair according to the equation

$$SNR_{eff} = \sqrt{2}\ \left[\text{erf}^{-1}(1-2P_{\text{FA}}) - \text{erf}^{-1}(1-2P_{\text{D}})\right]\ , \tag{1}$$

where the function $\text{erf}^{-1}(x)$ is the inverse of the standard error function, erf(x), which is defined by the equation

$$\text{erf(x)} = (2/\pi)\int_0^x dt\ \exp(-t^2)\ . \tag{2}$$

The basis for writing Eq. (1) for the (Gaussian equivalent) or SNR_{eff}, lies in the consideration that if we were dealing with a signal of magnitude s in Gaussian noise with standard deviation σ for which we can consider that

$$SNR_{eff} = s/\sigma\ , \tag{3}$$

then if the detection threshold were set at level T we would have

$$P_{\text{FA}} = 1 - 2/\pi \int_0^{T/(\sigma\sqrt{2})} dt\ \exp\left(-t^2\right)\ , \tag{4}$$

and

$$P_D = \tfrac{1}{2} - (2/\pi)\int_0^{(T-s)/(\sigma\sqrt{2})} dt\ \exp\left(-t^2\right)\ . \tag{5}$$

We can derive Eq. (1) from Eq.'s (3), (4), and (5).

Previously when discussing the PRBPRB graphs we approximated the SNR_{eff} as 3.0 for a (P_{D}, P_{FA})-pair of (0.92,0.08). We can use this pair and Eq. (1) to analytically calculate SNR_{eff}= 2.8101. We can then see that our approximation is close to the analytic value and suitable for the performance comparisons we will make between machine and operator.

II. EXPERIMENTAL APPROACH

We conducted a number of experiments using three types of targets and two types of backgrounds comparing operator performance with that provided by various types of filters. The three target types are referred to as "Transparent," "Uniform-Opaque," and "Camouflaged-Opaque." The two background types were "Random White Gaussian (RWG)" and "SPOT" patterns.

All the targets were disk shaped and nominally uniform across the disk. (We shaped the targets as closely as possible to disks given the square pixel array their patterns were fit onto.) Targets had some particular level of signal strength above the local average background. This excess of the signal strength above the local average background we shall refer to as the target's signature.

For the Transparent target the background pattern was altered only by adding the target signature to the background value of each pixel "covered" by the target disk.

For the Uniform-Opaque target each pixel covered by the disc was given the same value, a value equal to the target signature plus the average of all the background pixel values covered by the disk.

For the Camouflaged-Opaque target we used the same Uniform-Opaque target, but then also added a camouflage pattern on top of the uniform pattern. We obtained the camouflage pattern by randomly selecting a disk shaped region from somewhere else on the SPOT background image. From this region we extracted a pattern which was the same size as the target disk and introduced a bias to make the (camouflage) pattern thus extracted have a zero mean value. Examples of the three target types are shown in Fig.'s 14, 15, and 16. The association of these targets with real world target situations is discussed shortly.

We conducted experiments with (background clutter) noise either generated by the computer (in the case of white Gaussian noise) or obtained from satellite imagery of Jacksonville, Fla. (obtained from the French SPOT satellite system). We found that for the satellite imagery and Transparent target the optimum linear filter operates with an SNR_{eff} ranging from significantly higher than to just a bit better than the operator's — how much better depending on the target size. For the "Camouflaged Opaque" target in SPOT background the operator achieves a significantly higher SNR_{eff} than the machine/filter for all target sizes.

We generated the two types of backgrounds in the following manner. Random White Gaussian (RWG) backgrounds were generated by setting each pixel to the same value, a value between zero and one. We then added to the pixels a Gaussian random value with a standard deviation considerably less than unity. The choice of the random values added was statistically independent from pixel to

pixel. The results were then clipped to lie between zero and one. An example of RWG background is shown in Fig.11. The background pattern thus produced had no identifiable features.

SPOT images were generated by randomly selecting a small section of pixels from a large SPOT image of Jacksonville, Fla. area. Although the background clutter was random, features both man made and natural were clearly visible. These features had distinct edges. An example of the satellite imagery is shown in Fig. 12.

We used two different types of filters in the experiments. These were the matched filter and the optimum filter. In order to understand these filters the reader must be familiar with some signal processing terms. We next discuss the meaning of these terms. A discussion of SNR, Power Spectral Density (PSD), optimum filter, optimum linear filter, and matched filter follow.

The SNR is the ratio of the average of all measurements taken in a set of observations when there is a target present, minus the average of all the measurements taken when there is no target present (*i.e.*, the target signature), divided by the rms deviation of all the measurements taken when the target was always present (or the rms deviation when the target was always absent).

The use of the term SNR implies the use of some filter and of a sample selection process. Note that in defining the SNR we refer to measurements when describing both the signal (numerator) and the noise (denominator). In making a measurement some environmental effect (light, heat, temperature) interacts with a sensor to produce an array of data that is then input to a filter.

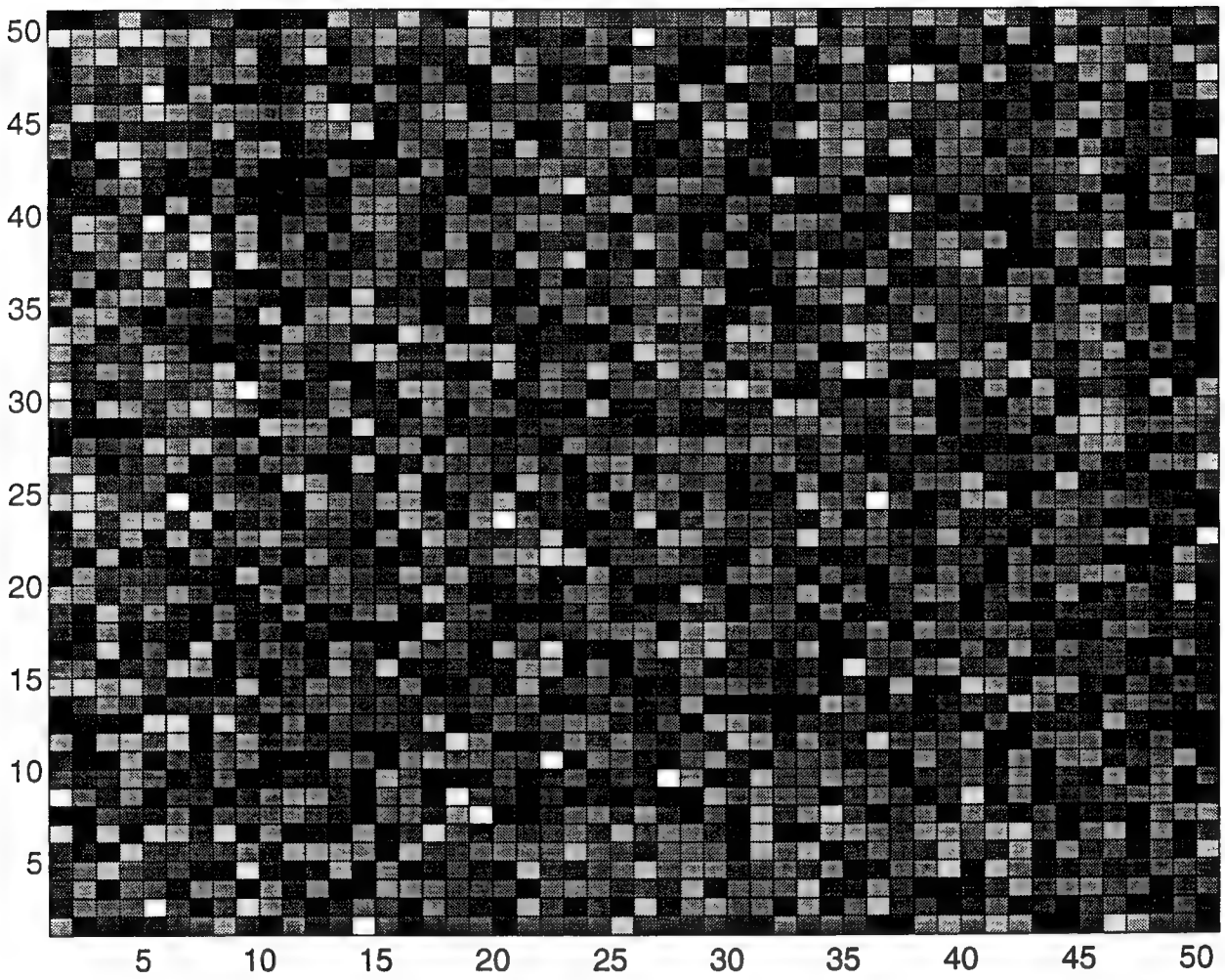


FIGURE 11. Random White Gaussian Background. The above figure shows an example of Random White Gaussian background. There is no target present in this figure. Note there are no identifiable patterns (features) in the background.

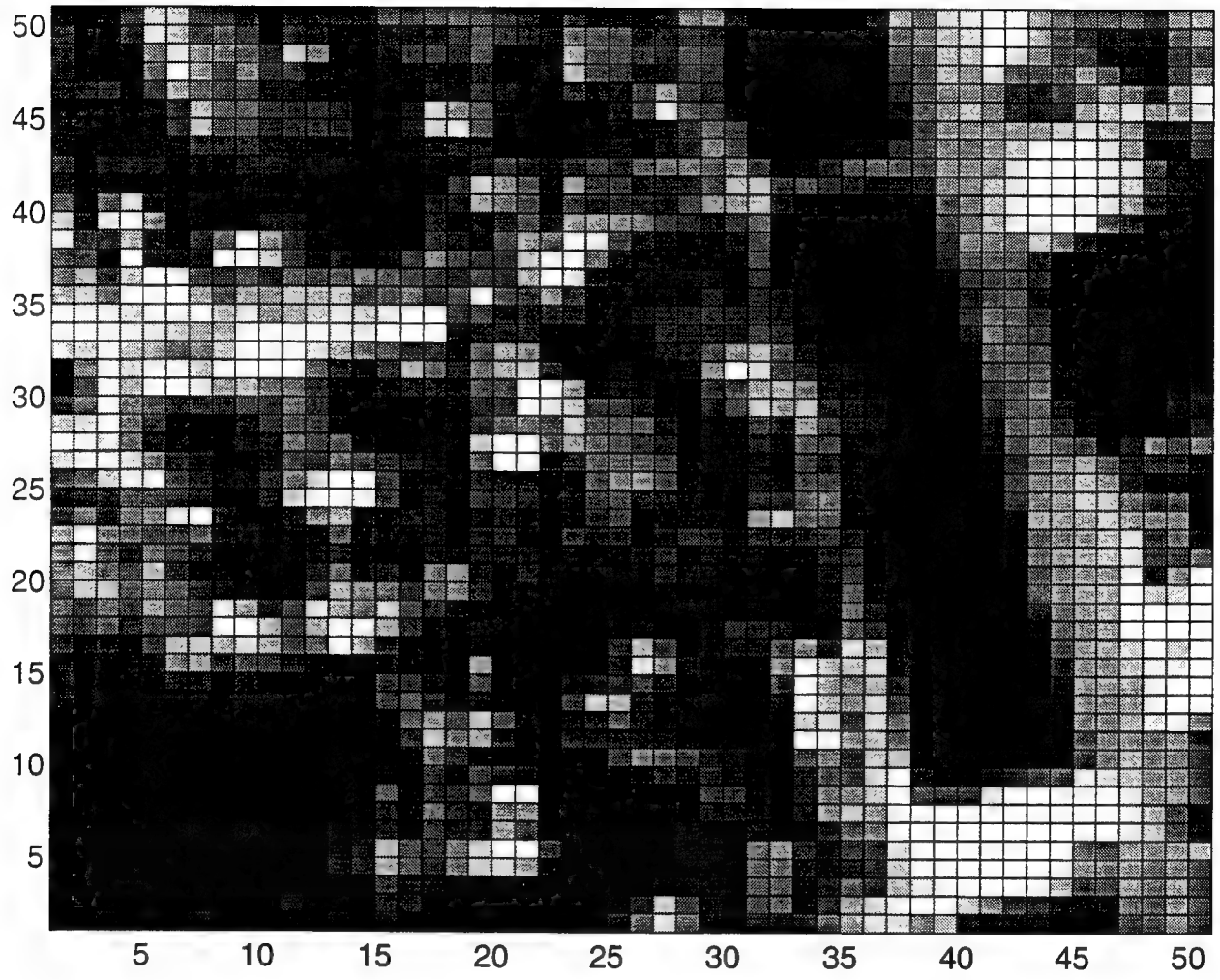


FIGURE 12. SPOT Background. The above figure shows a spot image. In this case there is no target present. Note there are clearly distinguished patterns (features) in the background. Linear filters can not take advantage of the nature of these features when deciding if a target is present or not present.

The filter may be thought of as some algorithm which processes a sequence of sensor output values to produce a sequence of filter output values. One output value for each possible source/target position with output sampling to correspond with some particular possible source/target position of the filter. Though filters are often thought of in terms of the processing of the various frequency components of the signal input to the filter it is at least as relevant to think of a filter as acting directly on each of the input sample values and it is in that sense that we shall generally consider a filter. If the filter is a linear filter then it is equivalent to the process of forming a sum of a set of weighting factors, each times one of the input signal sample values. (The sequence of filter output values corresponds to the forming of this sum with a sequence of different shifts between the weighting factors and the input signal sample values. Taking a single sample of the filter's output corresponds to restricting attention to a single relationship between the weighting factors and the input signal sample values.) A linear filter may be considered to be defined by it's set of weighting factors.

The PSD is the Fourier transform of the covariance of the background pixel values (nominally produced by a sensor). The PSD may be considered to define the frequency content of a random signal, in this case of the background, that will be part of the signal going into our filter.

The term matched filter refers to a linear filter which when expressed as a pattern of weighting factors may be considered to have the same "shape" as the target's nominal signal amplitude. In the two dimensional work of our thesis the matched filter for our circular target is a disk shaped pattern of equal weights, surrounded by a set of weights each equal to zero.

An optimum filter refers to a filter which when applied to a signal yields the maximum possible SNR. It may be linear or non-linear. In the case of RWG noise the optimum filter is known to be the matched filter.

An optimum linear filter is optimum only relative to the maximum SNR produced by all other linear filters. It is possible that some non-linear filter can produce a higher SNR than that produced by the optimum linear filter. For Gaussian noise that is not white the optimum linear filter is a linear filter whose weighting factors correspond to the matched filter Fourier transformed, divided by the PSD, and then inverse Fourier transformed.

We hypothesized that the human eye/brain combination would perform as an optimum non-linear filter.

In describing the experiments we conducted we will refer to various target types. This next

section discusses each of these targets in one of the previously mentioned backgrounds and describes the extent to which these target/background combinations may be considered to be related to real world target situations.

As already mentioned, the experiments used two types of background (RWG, SPOT) and three target types. The three target types are referred to as "Transparent," "Uniform-Opaque," and "Camouflaged-Opaque."

We combined these backgrounds and targets in three combinations each corresponding to one of the experiments. These combinations were:

1. Experiment 1. Transparent targets in Random White Gaussian (RWG) backgrounds.
2. Experiment 2. Transparent targets in SPOT images.
3. Experiment 3. Camouflaged-Opaque targets in SPOT images.

We considered conducting experiments with a fourth combination which was "Uniform-Opaque" targets in SPOT images but these targets were too easy for the operator to detect and we didn't believe the experiment would yield any insight to the non-linear filtering process we were investigating.

When we refer to Transparent targets we mean targets that added the target signature level to the background signal level for each pixel covered by the target disk. The Transparent targets were the most difficult targets for the operator to detect. In effect, the Transparent targets had perfect camouflage. They did not disrupt the background pattern; they only added a constant, the targets signature level, to the background pattern. An example of the Transparent target in both RWG, and a SPOT image (the first and second combination) is shown in Fig.'s 13 and 14.

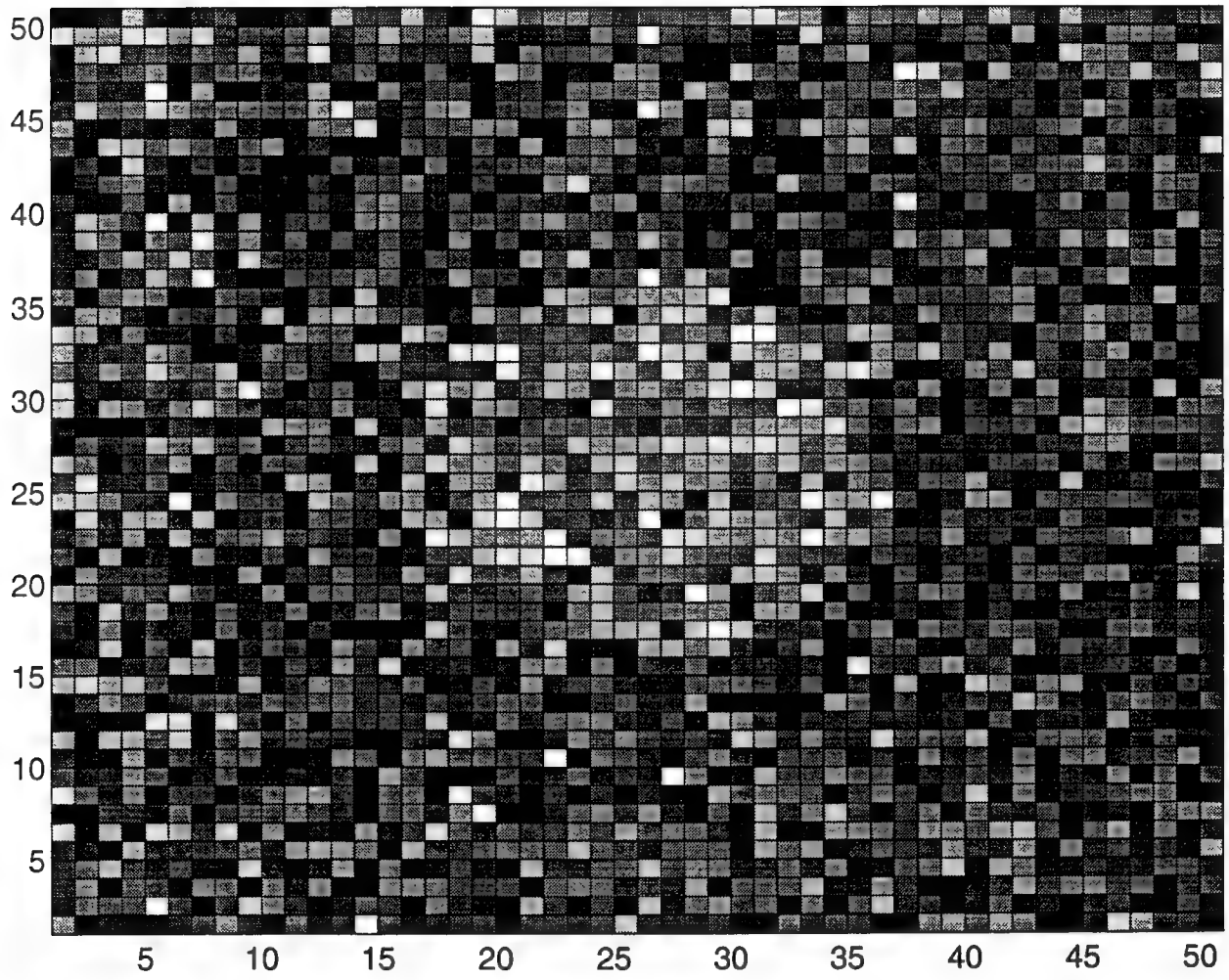


FIGURE 13. Random White Gaussian Background With A Transparent Target. The above figure shows random white Gaussian background with a target present. Look closely at the center of the image, the target has a diameter of 21 pixels.

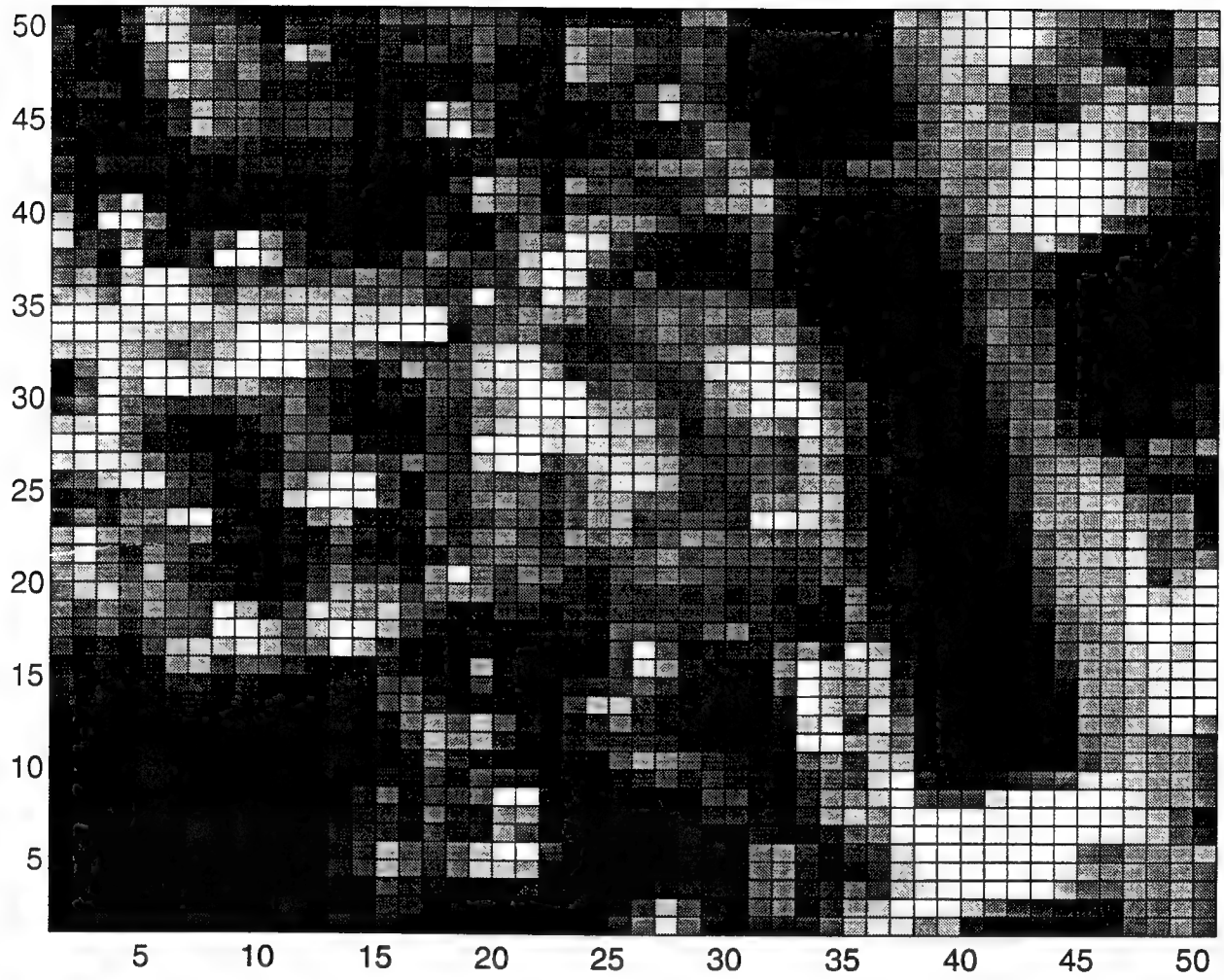


FIGURE 14. SPOT Image With A Transparent Target. The above figure shows a SPOT image with a target present. Look closely at the center of the image. The target which has diameterter of 21 pixels is just barely perceptable.

When we refer to either of the opaque class of targets we mean targets which, when displayed, had a uniform signal level. For each pixel covered by the target the target's signal (the same value for all target pixels) was substituted for the background. This created the effect of a uniform uncamouflaged target region that was very clearly not part of the background though the target pixel value was set to be only just a bit above (or in some cases not at all above) the average of the background pixel values covered by the target. The operator could recognize background patterns and features and easily decide that the "Uniform-Opaque" target feature was distinctly different from these background features. Since the "Uniform-Opaque" target was very easy for the operator to detect we did not do any quantitative work with it. An example of the Uniform Opaque target in a SPOT image is shown in Fig.15.

When we speak of camouflaged targets we refer to targets that are closely matched to but are imperfectly camouflaged relative to the Transparent targets. We created the imperfect camouflage by generating targets with a uniform signal level as we did with the Uniform Opaque target. We then randomly selected a target-size camouflage pattern from somewhere other than the the immediate vicinity of the target position and added that camouflage pattern to the uniform target pattern. We biased the camouflage pattern so that it had a zero mean signal value — so it did not change the target's average signature. The net effect was a uniform signal target with a nonuniform, zero-mean camouflage pattern that was statistically a good match to the background, but did not "join up" to the background very well at the target edges. An example of the Camouflaged-Opaque target in a SPOT image is shown in Fig.16.

We formulated each of these three types of target patterns to correspond (in a loose sense) to certain aspects of real world target situations. A description of one example of a real world target situation with characteristics corresponding to each target follows.

The Transparent target could represent a military vehicle (i.e., a tank) giving off an infrared (IR) signal. The tank will absorb the IR energy until its temperature equals the mean value of the background temperature. That is the tank absorbs and emits energy until it's temperature reaches equilibrium with the surrounding environment. When the tank has some internal heat source (i.e., an engine or a space heater) this heat source produces additional IR energy which causes the tank's IR signature to be above the IR signal produced by the mean background temperature.

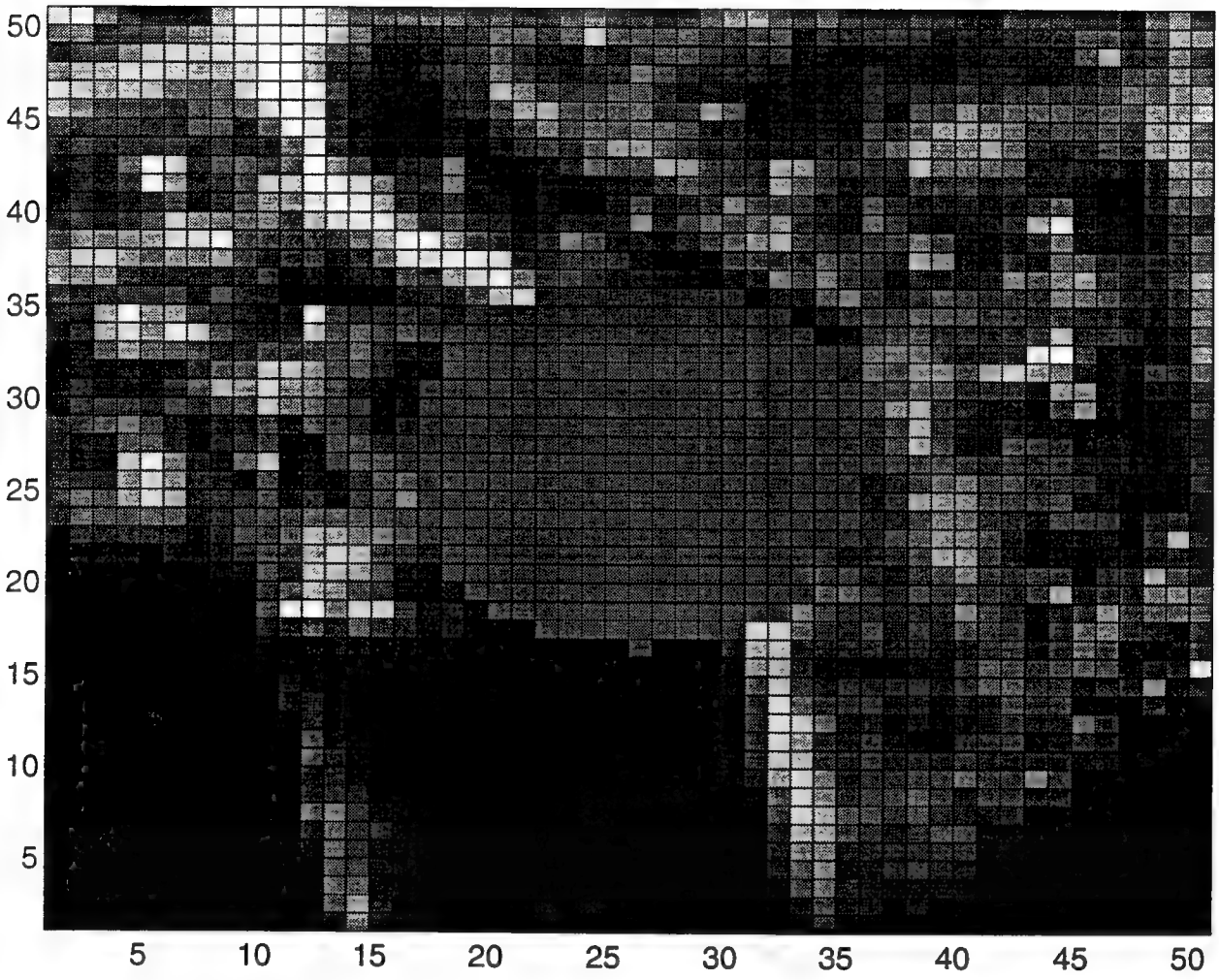


FIGURE 15. SPOT Image With Opaque Target. Shown above is a SPOT image with an opaque target. These targets were very easy to recognize because the target completely covered the background irregularities. We did not do any work with these types of targets.

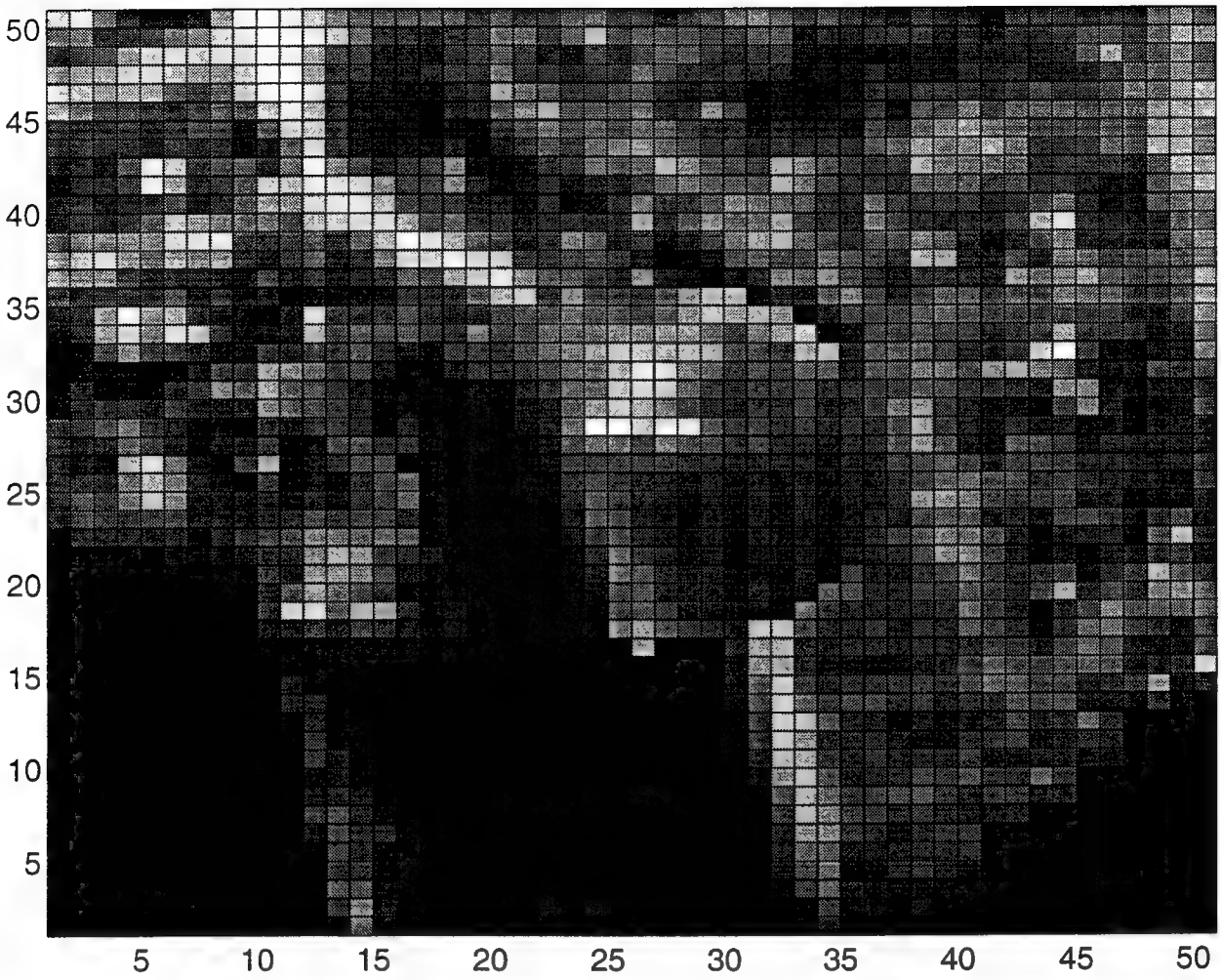


FIGURE 16. SPOT Image With A Camouflaged Opaque Target. The above figure shows a SPOT image with a Camouflaged Opaque Target. Look closely at the center of the background and find the target (it is difficult to see) then work your way out to the target edges. You will note that the edges don't quite 'match up' with the background.

Finally if the tank has a reasonably smart crew then they will not park the tank in the open, but seek out some natural camouflage. The natural camouflage if it's like light shrubbery is not solid, it also produces IR energy which is added to the overall target signal. The natural camouflage is not uniform; therefore, the target will not appear to have a uniform IR signal. Moreover, the nonuniformities have a pattern similar to the background. Each of the IR sources (tank, mean background, and natural camouflage) are essentially cumulative. The Transparent target in our experiment treats the various background signals and target signal cumulatively. Thus, the Transparent targets in a sense simulate many real world target detection situations where the targets have "near perfect" camouflage, (i.e., camouflage that blends into the background with no discontinuities).

Opaque targets simulate a target that has a uniform signal level. An uncamouflaged vehicle with a uniform IR emission is an example. For instance a tank in an open field with its hot engine off long enough for a uniform temperature distribution to have developed somewhat above that of the background. When we displayed the opaque targets to the operator we found that he could readily identify them. The operator was able to do this by recognizing the target's distinct shape and the absence of the irregularities characteristic of the background pattern.

The opaque target itself does not adequately simulate real world situations since it is unlikely to find military vehicles uncamouflaged. However, it is not unlikely to find military vehicles imperfectly camouflaged. We recognized that the Transparent targets represented the best possible camouflage and the Uniform Opaque targets represented the worst possible camouflage. We therefore decided to use a Uniform Opaque target camouflaged with a target sized background pattern in an attempt to simulate the case of a well but imperfectly camouflaged target. Again using the military vehicle example, consider a tank with a hot engine. The tank crew attempts to camouflage using camouflage paint, netting and some type of material to break the outline of the vehicle (such as dead foliage, hay or burlap). The crew positions the tank in some naturally concealed position. It is unlikely that the crew can find a position such that the natural IR signal of the background perfectly matches the IR signal of the tank. The tank absorbs the background heat from the environment and emits heat from its engine. The camouflage paint, netting and foliage provide pattern irregularities that help the tank to blend with the background. The ability of the crew to successfully match the background IR pattern depends on many factors (i.e., crew skill, natural camouflage, preparation time, etc). The best the crew can do is nearly perfectly match the background.

If the crew imperfectly camouflages the vehicle then the IR signal given off by the tank appears

as the uniform IR energy from the tank's engine, the mean background IR energy, and the IR energy from the camouflage. The crew's camouflage attempts break up the uniform signal but can not completely disguise it. The camouflage adds additional non-uniform signal which is close but imperfectly matched to the background IR pattern. The above mentioned "Opaque-Camouflaged" targets represent this case. In the "Opaque Camouflaged" situation we randomly selected the target camouflage pattern from some place in the background. Sometimes this camouflage matched the background nearly perfectly (a good crew) and sometimes it did not (a poorly trained crew). We thought this situation fairly represented an imperfectly camouflaged target.

In the previous sections we discussed the terminology and data presentation techniques the reader needs to understand the experiments. This terminology will be used to describe the experiments and the associated theory. The next chapter discusses some of this relevant theory.

III. SUPPORTING THEORY

This section introduces the theory relevant to the research. We will discuss the theory we consider essential to understanding the key aspects of the thesis. This section includes a discussion of:

1. Statistically reliable experiments and the number of trials required.
2. The definition and properties of linear filters.
3. Maximum SNR and optimum linear filtering.

A. STATISTICALLY RELIABLE EXPERIMENTS

The first point of theory discusses the process we used in determining the number of trials* necessary for each experiment to reliably determine probability of detection, probability of false alarm and SNR_{eff}. We wanted to make sure we ran enough trials in each experiment so that when we calculated a value for SNR_{eff} the rms error associated with this value would not be large relative to the value itself and thus make the result unreliable. We decided that the rms error should be no more than about fifteen percent of the "correct" answer.

We evaluated the relationship between number of trials and the standard deviation of the experiment's results for the SNR_{eff} using Monte Carlo techniques. In this Monte Carlo process we replaced the detailed procedure for each trial of determining the presence of a target with a randomly made choice of whether or not a mistake was made. The random choice was made in a way that reflected some expected (P_{D},P_{FA})-pair. The choice of this expected (P_{D},P_{FA})-pair was somewhat arbitrary and reflected our desire to achieve a convenient to work with detection probability and false alarm probability (i.e., values not too close to zero or one). The (P_{D},P_{FA}) values that we used where chosen to correspond to a SNR between one and five.

We then set up a simulation based on some number of trials which we considered using in an experiment. A result was randomly chosen for each of these trials. This result was either chosen to be correct or incorrect in proportion with the selected probabilities (P_{D},P_{FA}). The program first took the set of reports from the simulated trials, then calculated the experiment's Monte Carlo (P_{D},P_{FA})-pair — and from that calculated an SNR_{eff} for a simulated experiment.

That simulated experiment was repeated many times and from the results of those many trials we obtained a number of SNR_{eff} estimates. From these many SNR_{eff} estimates we determined a standard deviation to be associated with a SNR estimate. Having decided that an rms deviation

* Note when we use the word trial we mean a single detection event.

of 15 percent of the SNR_{eff} was a satisfactory result, if the standard deviation obtained with the simulation was not within 15 percent of the SNR_{eff} we repeated the experiment with an increased number of trials in the Monte Carlo simulation.

We repeated this process for different numbers of trials (50 trials to 500 trials) with (P_{D},P_{FA})-pairs chosen to correspond to SNR's of 1,2,3,4,5. In doing this we were conscious of the fact that if we asked the operator to perform too many trials, then human factors (boredom and annoyance) might unduly influence the experiment. We knew from experience that the operator could perform about 40-60 trials in an hour and expected that the longest the operator would be able to maintain his concentration was about eight hours. We therefore felt that 500 hundred trials was the absolute maximum number of operator trials for any experiment. We never ran an evaluation for a larger number of trials.

Shown in Table 1 is a summary of the Monte Carlo simulation results. We decided to use a SNR_{eff} of 3 and 200 experimental trials. We accepted an rms error of 14 percent.

The results shown in Table 1 indicate that for a nominal SNR of 5.0 the estimated SNR tends to be biased negatively, more so the smaller the number of trials. Even for a nominal SNR of 4.0 this bias was present at a significant level unless we used more than 100 trials. For our choice of a nominal SNR of 3.0 with 200 trials there does not appear to be any bias problem, We also note from consideration of Table 1 that for 200 trials we will have close to 15 percent error whether the nominal SNR is 2.0,3.0,4.0 – so we do not have to be too concerned about the exact selection of signal strength to match the nominal SNR value of 3.0.

We concluded from this that we could obtain satisfactory results if we ran two hundred trials in each operator experiment. (The two hundred trial figure was used only for the experiments with the human. For the experiment with the machine/filter we had no comparable constraint on how many trials we could run. For the machine/filter part of each experiment we ran ten thousand trials.)

In setting up the actual experiments we could not explicitly choose the SNR to match that in the table. Instead we simply guessed at a suitable signal strength and ran a small number of trials with a human and collected (P_{D},P_{FA}) data from which a coarse estimate of SNR_{eff} was found. We then accordingly adjusted the signal strength and repeated the experiment. After a few experiments we converged on a suitable signal strength. This signal strength was then used for both the two hundred operator trials and the ten thousand trials for the machine/filter.

SNR_{nom}	SNR_{eff}	STDSNR	#trials	PD_{eff}	PFA_{eff}	% dev
1.0000	1.0417	0.5509	50.0000	0.6920	0.3080	55.0905
2.0000	2.0860	0.6225	50.0000	0.8404	0.1596	31.1245
3.0000	2.8879	0.5519	50.0000	0.9178	0.0822	18.3974
4.0000	3.3172	0.3439	50.0000	0.9487	0.0513	8.5978
5.0000	3.4451	0.1951	50.0000	0.9567	0.0433	3.9022
1.0000	1.0142	0.3772	100.0000	0.6908	0.3092	37.7221
2.0000	2.0521	0.4536	100.0000	0.8415	0.1585	22.6817
3.0000	3.0739	0.5447	100.0000	0.9311	0.0689	18.1567
4.0000	3.7504	0.4202	100.0000	0.9666	0.0334	10.5039
5.0000	4.0113	0.2393	100.0000	0.9767	0.0233	4.7862
1.0000	1.0123	0.3108	150.0000	0.6915	0.3085	31.0778
2.0000	2.0330	0.3637	150.0000	0.8414	0.1586	18.1870
3.0000	3.0738	0.4794	150.0000	0.9327	0.0673	15.9787
4.0000	3.9326	0.4518	150.0000	0.9724	0.0276	11.2950
5.0000	4.3077	0.2558	150.0000	0.9836	0.0164	5.1161
1.0000	1.0072	0.2657	200.0000	0.6912	0.3088	26.5662
2.0000	2.0252	0.3134	200.0000	0.8414	0.1586	15.6719
3.0000	3.0644	0.4137	200.0000	0.9334	0.0666	13.7892*
4.0000	4.0062	0.4681	200.0000	0.9744	0.0256	11.7036
5.0000	4.4853	0.2862	200.0000	0.9867	0.0133	5.7244
1.0000	1.0060	0.2372	250.0000	0.6913	0.3087	23.7238
2.0000	2.0263	0.2751	250.0000	0.8422	0.1578	13.7526
3.0000	3.0521	0.3693	250.0000	0.9334	0.0666	12.3096
4.0000	4.0453	0.4610	250.0000	0.9757	0.0243	11.5250
5.0000	4.6112	0.3079	250.0000	0.9886	0.0114	6.1584
1.0000	1.0033	0.2187	300.0000	0.6910	0.3090	21.8702
2.0000	2.0190	0.2529	300.0000	0.8417	0.1583	12.6474
3.0000	3.0428	0.3308	300.0000	0.9334	0.0666	11.0253
4.0000	4.0712	0.4575	300.0000	0.976	0.0235	11.4387
5.0000	4.7057	0.3290	300.0000	0.9898	0.0102	6.5793
1.0000	1.0022	0.1997	350.0000	0.6910	0.3090	19.9726
2.0000	2.0138	0.2334	350.0000	0.8414	0.1586	11.6711
3.0000	3.0321	0.3038	350.0000	0.9331	0.0669	10.1254
4.0000	4.0701	0.4427	350.0000	0.9767	0.0233	11.0664
5.0000	4.7788	0.3402	350.0000	0.9907	0.0093	6.8031
1.0000	1.0031	0.1858	400.0000	0.6913	0.3087	18.5840
2.0000	2.0133	0.2193	400.0000	0.8415	0.1585	10.9668
3.0000	3.0274	0.2798	400.0000	0.9331	0.0669	9.3251
4.0000	4.0779	0.4240	400.0000	0.9771	0.0229	10.5998
5.0000	4.8370	0.3527	400.0000	0.9913	0.0087	7.0535
1.0000	1.0040	0.1752	450.0000	0.6915	0.3085	17.5158
2.0000	2.0099	0.2051	450.0000	0.8413	0.1587	10.2564
3.0000	3.0230	0.2682	450.0000	0.9330	0.0670	8.9388
4.0000	4.0690	0.4023	450.0000	0.9771	0.0229	10.0564
5.0000	4.8782	0.3633	450.0000	0.9918	0.0082	7.2651
1.0000	1.0018	0.1656	500.0000	0.6912	0.3088	16.5635
2.0000	2.0097	0.1922	500.0000	0.8414	0.1586	9.6123
3.0000	3.0226	0.2561	500.0000	0.9331	0.0669	8.5362
4.0000	4.0631	0.3864	500.0000	0.9771	0.0229	9.6611
5.0000	4.9177	0.3692	500.0000	0.9922	0.0078	7.3837

Table 1. Table Of Monte Carlo Simulations. The above figure shows the relationship between the nominal signal-to-noise ratio (SNR_{nom}), the effective signal-to-noise-ratio (SNR_{eff}), the deviation of snrnom (STDSNR), the number of trials (# trials), the effective PD (PD_{eff}), the effective PFA (PFA_{eff}), and the standard deviation expressed as a percent (% dev). In our experiments we used SNR_{nom}=3.0, and 200 trials (this row is marked with an asterisk). We accepted about a 14% standard deviation in the estimate of SNR_{eff} that an experiment will produce.

B. THE LINEAR FILTER

The second point in the theory that I will discuss concerns the concept of a linear filter. The basic property of a linear filter is that when it operates on two signals added together it produces a result equal to the sum of what it produces when acting on each signal separately. This is shown below with $F(...)$ denoting the filter and X, Y denoting the two signals. If F is a linear filter then

$$F(X+Y) = F(X) + F(Y). \tag{1}$$

As a corollary we also have that if a is a scalar constant then

$$F(aX) = aF(X). \tag{2}$$

It may be seen from this that in linear filter processing of a signal the "significance" assigned to any portion of the input signal in forming the output can not depend on the value of any other portion of the input signal. (This follows from the recognition that we may consider the two portions of the input signal to constitute two separate signals X and Y which when added together as $X+Y$, form the actual input. Clearly the value of one, say Y, cannot influence how we treat the other value, namely X.) It thus follows that a linear filter's output may be expressed as a weighted sum of the input values. If the input signal is a sequence of sample values, X_i where i is $(...-2,-1,0,1,2,...)$ then the sequence of filter output sample values , $\overline{X}_k$, may be written as

$$\overline{X}_k = \sum_{j=-\infty}^{\infty} F_j X_{k-j}. \tag{3}$$

where the set of values, F_j, is the set of weighting factors which define the filter.

Going from the discrete to the continuous we can then write that if the input signal is a function of some variable, say of time t, and so is expressible as $X(t)$,then the continuous output of the filter, $\overline{X}(t)$, is given by the expression

$$\overline{X} = \int_{-\infty}^{\infty} dt'\ F(t')X(t-t'). \tag{4}$$

This is the spatial domain (or time domain) **convolution** version of the linear filter process. Eq. (4) represents the convolution of the filter function, $F(t)$, with the function $X(t)$.

With suitable use of Fourier transforms an alternative form of the filtering process can be written.

$$\widetilde{F}(f) = \int_{-\infty}^{\infty} dt\ \exp(-2\pi i f t)\, F(t),$$

$$\widetilde{X}(f) = \int_{-\infty}^{\infty} dt\ \exp(-2\pi i f t)\, X(t),$$
$$F(t) = \int_{-\infty}^{\infty} df\ \exp(2\pi i f t)\, \widetilde{F}(f),$$
$$X(t) = \int_{-\infty}^{\infty} df\ \exp(2\pi i f t)\, \widetilde{X}(f). \tag{5}$$

Eqs. (5) are the mathematical expressions for the Fourier transforms and the inverse Fourier transforms in the time and frequency domains, respectively . We use the tilde notation, as in $\widetilde{X}(f)$, to denote the Fourier transform of the function, in this case $X(t)$. The alternative form of the filtering process is derived next.

The repeated Fourier transform is known to have the property that for any well-behaved function $G(p)$ we can write

$$G(p) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dp'\, dq \exp(\pm 2\pi i q(p-p'))G(p'). \tag{6}$$

We will use Eq. (6) in the derivation without offering proof. Replacing $F(t)$ and $X(t-t')$ with the associated Fourier transforms in Eq. (4) we get

$$\begin{aligned}
\overline{X}(t) &= \int_{-\infty}^{\infty} dt' \int_{-\infty}^{\infty} df\ \exp(2\pi i f t') \int_{-\infty}^{\infty} df' \exp(2\pi i f'(t-t'))\widetilde{F}(f)\widetilde{X}(f') \\
&= \int_{-\infty}^{\infty} df \int_{-\infty}^{\infty} df' \int_{-\infty}^{\infty} dt'\ \exp(2\pi i(f't - f't' + ft'))\widetilde{F}(f)\widetilde{X}(f') \\
&= \int_{-\infty}^{\infty} df \int_{-\infty}^{\infty} df' \int_{-\infty}^{\infty} dt'\ \exp(2\pi i t'(f-f'))\exp(-2\pi i f' t)\widetilde{F}(f)\widetilde{X}(f') \\
&= \int_{-\infty}^{\infty} df\, \widetilde{F}(f)\widetilde{X}(f)\exp(2\pi i f t).
\end{aligned} \tag{7}$$

Eq. (7) is the **frequency domain** version of the filter process for a linear filter, where $\widetilde{F}(f)$ represents an alternate form of definition of the filter function, **the frequency domain** version of the filter function. It is equal to the Fourier transform of the weighting factor or **convolution** version of the filter function, $F(t)$.

C. THE OPTIMUM LINEAR FILTER

This next section discusses the optimum linear filter. An optimum linear filter operates on a signal in such a manner as to produce the highest possible SNR that any linear filter can provide.*

It is our aim in this section to develop an expression for the optimum linear filter. For simplicity we will develop the expression in one dimension. The extension of this proof to two dimensions is straight forward; it only requires a bit more algebra.

We intend to represent the signal as $S(t)$. Both the target and the background contribute to the total signal such that

$$S(t) = s(t) + n(t). \tag{1}$$

Here $s(t)$ is the target signature and $n(t)$ is the background signal. $n(t)$ can be considered to be equal to a random function of unit rms amplitude, $\nu(t)$, multiplied by a scaling constant, N_0, so we have

$$n(t) = N_0\,\nu(t). \tag{2}$$

The function $\nu(t)$ has the following properties and associated quantities.

---

* In order to organize nomenclature for the reader's easy reference we list here the terms required for this proof.

$P(t)$ = One dimensional representation that describes a target,
$\widetilde{P}(f)$ = Fourier transform of $P(t)$,
$s(t) = S_0\,P(t - t_0)$; Target signature,
$\widetilde{s}(f)$ = Fourier transform of $s(t)$,
$n(t)$ = Noise associated with the target signal,
$\widetilde{n}(f)$ = Fourier transform of $n(t)$,
$F(t)$ = Linear filter,
$\widetilde{F}(f)$ = Fourier transform of $F(t)$,
$F_{op}(t)$ = A linear filter which we will prove is optimum,
$\widetilde{F}_{op}(f)$ = Fourier transform of $F_{op}(t)$,
$\delta F(t)$ = Filter that consists of functions orthogonal to $F_{op}(t)$,
$\delta\widetilde{F}(f)$ = Fourier transform of $\delta F(t)$,
$\overline{S}(t)$ = Filtered background & target signal in the time domain,
$\overline{s}(t)$ = Filtered target signal in the time domain,
$\overline{n}(t)$ = Filtered background noise signal in the time domain,
$\nu(t)$ = Function of unit strength which describes the background characteristics,
$\widetilde{\nu}(f)$ = Fourier transform of $\nu(t)$,
N_0 = Weighting factor which when multiplied by $\nu(t)$ describes the background,
$C_\nu(t)$ = Covariance of the background signal,
$\widetilde{\Phi}_\nu(f)$ = Power spectral density.

1. Its ensemble average is zero, so we can write

$$\langle \nu(t) \rangle = 0. \tag{3}$$

From this it can be seen that

$$\langle n(t) \rangle = 0. \tag{4}$$

2. Its Fourier transform is expressed as

$$\widetilde{\nu}(f) = \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, \nu(t) \; . \tag{5}$$

so that

$$\widetilde{n}(f) = \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, n(t). \tag{6}$$

3. It's characterized by the correlation function with

$$C_\nu(t) = \langle \nu(t - t') \nu(t') \rangle. \tag{7}$$

4. It has a power spectral density, $\widetilde{\Phi}_\nu(f)$, which is the Fourier transform of $C_\nu(t)$, so that

$$\widetilde{\Phi}_\nu(f) = \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, C_\nu(t). \tag{8}$$

We will let

$$s(t) = S_0 \, P(t - t_0), \tag{9}$$

represent the signature of the target. Here P(t) describes the normalized target signal form. It's normalization is expressed as

$$\int_{-\infty}^{\infty} dt \, |P(t)|^2 = 1. \tag{10}$$

It's Fourier transform is

$$\widetilde{P}(f) = \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, P(t). \tag{11}$$

The quantity S_0 may be considered to be a measure of the target signal strength, and t_o may be considered to define the target's "location/position" , or time of occurrence. The frequency transform of the signal can be written

$$\begin{aligned}
\widetilde{s}(f) &= \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, s(t) \\
&= \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, S_0 \, P(t - t_0) \\
&= S_0 \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, P(t - t_0) \\
&= S_0 \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, P(t - t_0) \, \exp(2\pi i f t_0) \exp(-2\pi i f t_0) \\
&= S_0 \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f (t - t_0)) \, P(t - t_0) \, \exp(-2\pi i f t_0) \; .
\end{aligned} \tag{12}$$

We shall let $t' = (t - t_0)$, and with that we have dt'=dt. This allows us to write

$$\begin{aligned}\widetilde{s}(f) &= S_0 \exp(-2\pi i f t_0) \int_{-\infty}^{\infty} dt' \, \exp(-2\pi i f t') P(t') \\ &= S_0 \exp(-2\pi i f t_0) \, \widetilde{P}(f). \end{aligned} \tag{13}$$

We can now write

$$\begin{aligned}\widetilde{S}(f) &= \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, S(t) \\ &= \int_{-\infty}^{\infty} dt \, \exp(-2\pi i f t) \, [s(t) + n(t)] \\ &= \widetilde{s}(f) + \widetilde{n}(f). \end{aligned} \tag{14}$$

We have expressions for the background and target signal Eqs. (2), (6), (9), and (13) — and the associated Fourier transforms. We next develop an expression for filtered signal. The filtered signal, which we denote by $\overline{S}(t)$, can be written in accordance with Eq. (14) as

$$\begin{aligned}\overline{S}(t) &= \int_{-\infty}^{\infty} df \, \exp(2\pi i f t) \, \widetilde{F}(f) \widetilde{S}(f) \\ &= \int_{-\infty}^{\infty} df \, \exp(2\pi i f t) \, \widetilde{F}(f) [\widetilde{s}(f) + \widetilde{n}(f)] \\ &= \int_{-\infty}^{\infty} df \, \exp(2\pi i f t) \, \widetilde{F}(f) \widetilde{s}(f) + \int_{-\infty}^{\infty} df \, \exp(2\pi i f t) \, \widetilde{F}(f) \widetilde{n}(f). \end{aligned} \tag{15}$$

If we write:

$$\begin{aligned}\overline{s}(t) &= \int_{-\infty}^{\infty} df \, \exp(2\pi i f t) \, \widetilde{F}(f) \widetilde{s}(f), \\ \overline{n}(t) &= \int_{-\infty}^{\infty} df \, \exp(2\pi i f t) \, \widetilde{F}(f) \widetilde{n}(f), \end{aligned} \tag{16}$$

then we get

$$\overline{S}(t) = \overline{s}(t) + \overline{n}(t). \tag{17}$$

It is easy to see that the expected value of the filtered signal taken at time t_0 is

$$\overline{S}(t_0) = \overline{s}(t_0) + \overline{n}(t_0), \tag{18}$$

and from Eq. (4) have $\langle \overline{n}(t_0) \rangle = 0$, so

$$\begin{aligned}\langle \overline{S}(t_0) \rangle &= \langle \overline{s}(t_0) \rangle \\ &= \overline{s}(t_0). \end{aligned} \tag{19}$$

This allows us to write for the random noise

$$|\overline{n}(t_0)| = |\overline{S}(t_0) - \overline{s}(t_0)|. \tag{20}$$

so that the mean square, or noise power, is:

$$\langle |\overline{n}(t_0)|^2 \rangle = \langle |\overline{S}(t_0) - \overline{s}(t_0)|^2 \rangle. \tag{21}$$

Since the mean signal power is

$$\langle \overline{S}(t_0) \rangle^2 \tag{22}$$

the signal-to-noise power ratio, SNR_p, is:

$$\begin{aligned} SNR_p &= \frac{\langle \overline{S}(t_0) \rangle^2}{\langle |\overline{n}(t_0)|^2 \rangle} \\ &= \frac{[\overline{s}(t_0)]^2}{\langle |\overline{n}(t_0)|^2 \rangle}. \end{aligned} \tag{23}$$

We are now ready to consider the form of our filter. We work with a linear filter which, in frequency domain form we shall denote by $\widetilde{F}(f)$. Our objective is to choose a filter form which maximizes the signal-to-noise power ratio, $SNRp$. We intend to show that the optimum filter is of the form

$$\widetilde{F}_{op}(f) = \frac{\widetilde{P}^*(f)}{\widetilde{\Phi}_\nu(f)}. \tag{24}$$

But for the moment we simply condider this to be some "arbitrarily" chosen filter form with no established virtue. We will do our analysis considering a more general filter form, namely

$$\widetilde{F}(f) = \widetilde{F}_{op}(f) + \delta\widetilde{F}(f). \tag{25}$$

We intend to show that when $\delta\widetilde{F}(f) = 0$ the filter $\widetilde{F}(f)$ provides the largest possible value for $SNRp$. This will show that $\widetilde{F}_{op}(f)$ is the optimum filter.

Before proceeding any further with the analysis we need to take note that if $\widetilde{F}_{op}(f)$ has some particular virtue as a filter then so does $(1+\alpha)\widetilde{F}_{op}(f)$. [For the latter the output signal is larger by a factor of $(1+\alpha)$, but so is the rms noise in the output. Therefore the signal-to-noise-ratio is unchanged by the presence of the factor of $(1+\alpha)$]. This means that for our proof of the optimality of $\widetilde{F}_{op}(f)$ we can restrict the possible forms of $\delta\widetilde{F}(f)$ to functions that are not expressible as $\alpha\widetilde{F}_{op}(f)$. In order to impose this restriction in an analytic form it is convenient to consider an orthogonal set

of functions one of whose members is $\widetilde{F}_{op}(f)$, and then restrict $\delta\widetilde{F}(f)$ to some weighted sum of all the *other* functions in this orthonormal set. We shall choose the orthonormal set to be orthogonal on the domain $[-\infty, +\infty]$, with a weighting function $\widetilde{\Phi}_\nu(f)$ corresponding to the power spectral density of the noise.The orthogonality of any two functions in our set, can be expressed as

$$\int_{-\infty}^{\infty} df\, \widetilde{F}_p^*(f)\widetilde{F}_q(f)\widetilde{\Phi}_\nu(f) = \begin{cases} 1, & \text{if } p = q; \\ 0, & \text{if } p \neq q. \end{cases} \tag{26}$$

The orthogonality of $\widetilde{F}_{op}(f)$ with respect to all of the other functions in the set and the fact that $\delta\widetilde{F}(f)$ is formed as a weighted sum of all those other functions means

$$\int_{-\infty}^{\infty} df\, \widetilde{F}_{op}^*(f)\, \delta\widetilde{F}(f)\, \widetilde{\Phi}_\nu(f) = 0. \tag{27}$$

We can write the expression for the filtered signal at time t_0 as

$$\begin{aligned}
\overline{s}(t_0) &= \int_{-\infty}^{\infty} df\, \exp(2\pi i f t_0)\, \widetilde{F}_{op}(f)\, S_0 \exp(-2\pi i f t_0)\widetilde{P}(f) \,+\, \int_{-\infty}^{\infty} df\, \exp(2\pi i f t_0)\, \delta\widetilde{F}(f)\, S_0 \exp(-2\pi i f t_0)\widetilde{P}(f) \\
&= \int_{-\infty}^{\infty} df\, \exp(2\pi i f (t_0 - t_0))\, \widetilde{F}_{op}(f)\, S_0\widetilde{P}(f) \,+\, \int_{-\infty}^{\infty} df\, \exp(2\pi i f (t_0 - t_0))\, \delta\widetilde{F}(f)\, S_0\widetilde{P}(f) \\
&= \int_{-\infty}^{\infty} df\, \widetilde{F}_{op}(f)\, S_0\widetilde{P}(f) \,+\, \int_{-\infty}^{\infty} df\, \delta\widetilde{F}(f)\, S_0\widetilde{P}(f) \\
&= S_0\int_{-\infty}^{\infty} df\, \widetilde{F}_{op}(f)\widetilde{P}(f) \,+\, S_0\int_{-\infty}^{\infty} df\, \delta\widetilde{F}(f)\widetilde{P}(f)
\end{aligned} \tag{28}$$

and since

$$\widetilde{P}^*(f) = \widetilde{F}_{op}(f)\widetilde{\Phi}_\nu(f) \tag{29}$$

$$\widetilde{P}(f) = \widetilde{F}_{op}^*(f)\widetilde{\Phi}_\nu(f) \tag{30}$$

we can write

$$\overline{s}(t_0) = S_0\int_{-\infty}^{\infty} df\, \widetilde{F}_{op}^*(f)\, \widetilde{F}_{op}(f)\, \widetilde{\Phi}_\nu(f) \,+\, S_0\int_{-\infty}^{\infty} df\, \widetilde{F}_{op}^*(f)\, \delta\widetilde{F}(f)\, \widetilde{\Phi}_\nu(f)\ . \tag{31}$$

But remember the integral of the optimum linear filter and the combination of orthogonal filters is zero

$$\int_{-\infty}^{\infty} df\, \widetilde{F}_{op}^*(f)\delta\widetilde{F}(f)\widetilde{\Phi}_\nu(f) = 0. \tag{32}$$

Therefore:

$$\begin{aligned}
\overline{s}(t_0) =& S_0\int_{-\infty}^{\infty} df\, \widetilde{F}_{op}^*(f)\widetilde{F}_{op}(f)\widetilde{\Phi}_\nu(f) \,+\, 0 \\
=& S_0\int_{-\infty}^{\infty} df\, \frac{\widetilde{P}^*(f)\widetilde{P}(f)}{\widetilde{\Phi}_\nu(f)\,\widetilde{\Phi}_\nu(f)}\, \Phi_\nu(f) \\
=& S_0\int_{-\infty}^{\infty} df\, \frac{|\widetilde{P}(f)|^2}{\widetilde{\Phi}_\nu(f).}
\end{aligned} \tag{33}$$

This is the numerator in our expression for the signal-to-noise power ratio, SNR_p.

We next develop the expression for average filtered noise the denominator in the signal-to-noise power ratio expression. Starting from the expression for the noise power $\langle|\overline{n}(t_0)|^2\rangle$ given in Eq. (21), we can write

$$\begin{aligned}\langle|\overline{n}(t_0)|^2\rangle =& \Big\langle \Big| N_0 \int_{-\infty}^{\infty} df\ \exp(2\pi i f t_0)\, \widetilde{F}_{op}(f)\widetilde{\nu}(f) \\ &\qquad + N_0 \int_{-\infty}^{\infty} df\ \exp(2\pi i f t_0)\, \delta\widetilde{F}(f)\widetilde{\nu}(f)\Big|^2 \Big\rangle \\ =& \Big\langle \Big[N_0 \int_{-\infty}^{\infty} df\ \exp(2\pi i f t_0)\, \widetilde{F}_{op}(f)\widetilde{\nu}(f) \\ &\qquad + N_0 \int_{-\infty}^{\infty} df\ \exp(2\pi i f t_0)\, \delta\widetilde{F}(f)\widetilde{\nu}(f)\Big] \\ &\times \Big[N_0 \int_{-\infty}^{\infty} df'\ \exp(2\pi i f' t_0)\, \widetilde{F}_{op}(f')\widetilde{\nu}(f') \\ &\qquad + N_0 \int_{-\infty}^{\infty} df'\ \exp(2\pi i f' t_0)\, \delta\widetilde{F}(f')\widetilde{\nu}(f')\Big]^* \Big\rangle. \end{aligned} \tag{34}$$

Recalling the form of $\widetilde{F}_{op}(f)$ we have:

$$\widetilde{F}_{op}(f)\, \widetilde{\Phi}_\nu(f) = \widetilde{P}^*(f), \tag{35}$$

$$\widetilde{F}_{op}(f)^*\, \widetilde{\Phi}_\nu(f) = \widetilde{P}(f). \tag{36}$$

Substituting this into Eq. (33) we get

$$\begin{aligned}\langle|\overline{n}(t_0)|^2\rangle =& \Big\langle \Big[N_0 \int_{-\infty}^{\infty} df\ \exp(2\pi i f t_0)\, \frac{\widetilde{P}^*(f)}{\widetilde{\Phi}_\nu(f)}\, \widetilde{\nu}(f) \\ &\qquad + N_0 \int_{-\infty}^{\infty} df\ \exp(2\pi i f t_0)\, \delta\widetilde{F}(f)\, \widetilde{\nu}(f)\Big] \\ &\times \Big[N_0 \int_{-\infty}^{\infty} df'\ \exp(2\pi i f' t_0)\, \frac{\widetilde{P}^*(f')}{\widetilde{\Phi}_\nu(f')}\, \widetilde{\nu}(f') \\ &\qquad + N_0 \int_{-\infty}^{\infty} df'\ \exp(2\pi i f' t_0)\, \delta\widetilde{F}(f')\, \widetilde{\nu}(f')^2\Big]^* \Big\rangle. \end{aligned} \tag{37}$$

Making double integrals of the product of integrals we get.

$$\begin{aligned}\langle|\overline{n}(t_0)|^2\rangle =\ & {N_0}^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} df\, df'\ \exp\big(2\pi i(f-f')t_0\big)\, \frac{\widetilde{P}^*(f)\, \widetilde{P}(f')}{\widetilde{\Phi}_\nu(f)\, \widetilde{\Phi}_\nu(f')}\, \big\langle \widetilde{\nu}(f)\widetilde{\nu}^*(f')\big\rangle \\ &+ {N_0}^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} df\, df'\ \exp\big(2\pi i(f-f')t_0\big)\, \frac{\widetilde{P}^*(f)\, \delta\widetilde{F}^*(f')}{\widetilde{\Phi}_\nu(f)}\, \big\langle \widetilde{\nu}(f)\widetilde{\nu}^*(f')\big\rangle \\ &+ {N_0}^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} df\, df'\ \exp\big(2\pi i(f-f')t_0\big)\, \frac{\delta\widetilde{F}(f)\, \widetilde{P}(f')}{\widetilde{\Phi}_\nu(f')}\, \big\langle \widetilde{\nu}(f)\widetilde{\nu}^*(f')\big\rangle \\ &+ {N_0}^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} df\, df'\ \exp\big(2\pi i(f-f')t_0\big)\, \delta\widetilde{F}(f)\, \delta\widetilde{F}^*(f')\, \big\langle \widetilde{\nu}(f)\widetilde{\nu}^*(f')\big\rangle \end{aligned} \tag{38}$$

It is known that *

$$\int_{-\infty}^{\infty} dt\, C_\nu(t)\, \exp(-2\pi i f t)\; g(f) = \widetilde{\Phi}_\nu(f)\, g(f), \tag{39}$$

where $g(f)$ is any well behaved function. Eq. (39) is now applied, yielding the result that

$$\left\langle |\overline{n}(t_0)|^2 \right\rangle = {N_0}^2 \int_{-\infty}^{\infty} df\, \frac{|\widetilde{P}(f)|^2}{\widetilde{\Phi}_\nu(f)} + {N_0}^2 \int_{-\infty}^{\infty} df\, \widetilde{P}^*(f) \delta\widetilde{F}^*(f)$$

$$+ {N_0}^2 \int_{-\infty}^{\infty} df\, \widetilde{P}(f)\, \delta\widetilde{F}(f) + {N_0}^2 \int_{-\infty}^{\infty} df\, |\delta\widetilde{F}(f)|^2 \widetilde{\Phi}_\nu(f). \tag{40}$$

We shall now, again use Eq.'s (29) and (30) make this substitution then take advantage of the orthogonality of $\widetilde{F}_{op}(f)$ with respect to all other filters to determine that the middle two integrals in Eq. (40) zero. Thus the expression for filtered noise is.

$$\begin{aligned} \left\langle |\overline{n}(t_0)|^2 \right\rangle &= {N_0}^2 \int_{-\infty}^{\infty} df\, \frac{|\widetilde{P}(f)|^2}{\widetilde{\Phi}_\nu(f)} + 0 + 0 + {N_0}^2 \int_{-\infty}^{\infty} df\, |\delta\widetilde{F}(f)|^2 |\widetilde{\Phi}_\nu(f) \\ &= {N_0}^2 \int_{-\infty}^{\infty} df\, \frac{|\widetilde{P}(f)|^2}{\widetilde{\Phi}_\nu(f)} + {N_0}^2 \int_{-\infty}^{\infty} df\, |\delta\widetilde{F}(f)|^2 |\widetilde{\Phi}_\nu(f). \end{aligned} \tag{41}$$

We now have expressions for filtered signal and filtered noise and can express the signal-to-noise power ratio in terms of $\widetilde{F}_{op}(f)$ and the filters orthogonal to $\widetilde{F}_{op}(f)$.

The derived expression for the filtered signal-to-noise power ratio is:

$$\begin{aligned} SNR_p &= \frac{[\overline{s}(t_0)]^2}{\left\langle |\overline{n}(t_0)|^2 \right\rangle} \\ &= \frac{{S_0}^2 \left(\int_{-\infty}^{\infty} df\, \frac{|\widetilde{P}(f)|^2}{\widetilde{\Phi}_\nu(f)} \right)^2}{{N_0}^2 \int_{-\infty}^{\infty} df\, \frac{|\widetilde{P}(f)|^2}{\widetilde{\Phi}_\nu} + {N_0}^2 \int_{-\infty}^{\infty} df\, |\delta\widetilde{F}(f)|^2 \widetilde{\Phi}_\nu(f)}. \end{aligned} \tag{42}$$

For the signal-to-noise-ratio to be as large as possible the denominator in the above equation has to be as small as possible. The smallest possible value for the denominator occurs when the second term is zero.†

---

* D.L. Fried, "Spectral and Angular Covariance of Scintillation for Propagation in a Randomly Inhomogeneous Medium", Appl. Opt. **10**, 721-731 (1971); Appendix

† This is true because none of the terms in the denominator are negative;
$\widetilde{\Phi}_\nu(f)$ is everywhere positive,
$|\widetilde{P}(f)|^2$ is everywhere positive,
$|\delta\widetilde{F}(f)|^2$ is everywhere positive.

The second term is zero when $\delta\widetilde{F}(f) = 0$. Recall that we began this proof by examining a filter combination which was the sum of the filter $\widetilde{F}_{op}(f)$ and an orthogonal filter . We have just concluded that this filter combination yields the maximum signal-to-noise power ratio when the orthogonal filter is equal to zero. Therefore the filter combination which yields the maximum signal-to-noise power ratio and is the optimum filter must be $\widetilde{F}_{op}(f)$ alone, which we represent by writing

$$\widetilde{F}(f) = \widetilde{F}_{op}(f) + \delta\widetilde{F}(f) \quad \text{definition,} \tag{43}$$

$$\delta\widetilde{F}(f) = 0 \quad \text{to maximize } SNR_p\,, \tag{44}$$

$$\widetilde{F}(f) = \widetilde{F}_{op}(f) \quad \text{the optimum filter .} \tag{45}$$

This completes our proof.

IV. PRESENTATION OF EXPERIMENTS AND RESULTS

This section discusses the experiments and highlights the results. In three types of experiments we tested a human subject's performance and a machine's performance under a variety of target and background conditions. The three experiments are characterized as follows:

1. Experiment 1: Transparent target in Random White Gaussian (RWG) noise.
2. Experiment 2: Transparent target in SPOT background.
3. Experiment 3: Camouflaged-Opaque target in SPOT background.

All of the experiments used a UNIX computer to generate pixel representations of a roughly circular target in background noise. The programming language was MATLAB.

Each experiment consisted of two separate processes. The first process displayed pictures on the computer monitor and was designed to obtain operator input. The second process digitized image data and presented this data to the machine/filter, which applied the optimum linear filter. This second process was designed to test how a machine/filter (using the optimum linear filter) would perform. At the end of each process the computer program that was "responsible" for the conduct of the experiment obtained, and stored the operator or machine data and displayed this data on a PRBPRB graph for assessment of SNR_{eff}.

The operator process consisted of a series of statistically independent trials. For each trial the subject was told the nominal location of the target but not whether the target was actually present in that trial. Thus the operator did not have to search for the target; he only had to decide if it was present. In each trial the targets had a probability of one half of being present, but only the computer program conducting the experiment "knew" for certain whether or not the target was actually present in that trial. The operator was subjected to no time limit in deciding whether or not the target was present. In experiments one and two the targets were Transparent so as to achieve perfect blending of their pattern with the background pattern. In experiment three the targets were opaque and used a camouflage pattern that, while statistically equivalent to the blended background pattern, did not smoothly join with the surrounding background at the target edge.

In order to provide a more operator friendly image display we displayed the image in several forms, side-by-side, each with a different gray-scale rendering. We felt that showing more than one gray-scale rendering was appropriate since in many real world applications an operator could control the intensity and contrast of the display monitor. For instance a radar operator, or the person viewing a FLIR display, generally has control knobs to adjust the display's contrast and brightness. In our

experiments we showed eighteen different gray-scale versions of the target/background images. The same image data (background signal and target signal) was present in each of the eighteen pictures in the monitor display. Thus if a target was present in a given trial then each of the eighteen pictures contained target signal, or if the target was not present in a given trial then none of the eighteen pictures contained target signal. Each of the 18 display pictures used a different transformation to prepare the pixel values for display. In making his decision as to the target's presence the operator was free to consider any one (or even several) of these eighteen versions of the image. In the experiments the circular target region had a slightly greater signature value if the target was present than it had when the target were not present (unless the target had zero signature, as in Experiment 3). The targets thus appeared slightly brighter to the operator than the surrounding background.

After we obtained the operator data we ran the second process which we designed to measure the machine/filter's performance. We used the same background and target definitions/parameters for the machine/filter as we did for the operator. The machine/filter decided whether targets were present or not present based on the filter output compared with a threshold value. Results were obtained for a variety of threshold values (actually for 100 different threshold values). Just as the operator knew where the target would be if it was present we also allowed the machine to have this knowledge. Accordingly the filtered signal output was taken only for the single sample output corresponding to the target's nominal location. The target present/not present decision was made by the machine by applying a threshold to this filter output value.

The experiments recorded both the operator and machine data then plotted the $(P_{\mathrm{D}}, P_{\mathrm{FA}})$-pairs on a PRBPRB graph as discussed in Chapter I. Operator data was collected over 200 trials leading to a fourteen percent standard deviation in the SNR_{eff}. Since machine data could be collected more rapidly, it was collected over 10,000 trials. Each $(P_{\mathrm{D}}, P_{\mathrm{FA}})$-pair plotted was computed using results from 10,000 trials for the machine/filter or 200 trials for the operator.

A. EXPERIMENT 1: TRANSPARENT TARGETS IN RANDOM WHITE GAUSSIAN NOISE

This experiment used Transparent targets in Gaussian random background noise. The target region had a slightly greater signal value if the target was present than when the target was not present. If the target was present it was greater by the target signature value. In the random background pattern this signature was not always enough to provide a clear indication of the target's presence. An example of the monitor display shown to the operator is enclosed in Appendix D.

We found that the machine achieved a SNR_{eff} near the theoretically predicted value for a linear filter. The operator very closely matched the machine/filter's performance. The operator performance in this experiment seemed to lend credibility to our belief that the operator can act as an optimum (non-linear) filter (see Fig. 17). These experimental results were not surprising for the machine/filter since a Gaussian "white" background is an ideal case for the optimum linear filter. In this case theory predicts that the machine/filter performance will be optimal and should correspond to the GLINE of appropriate signal strength. The best we expect the operator to do is match the machine/filter's performance. We were somewhat surprised by how close the operator performance did match the machine/filter's performance. As can be seen in Fig. 17 the operator's performance was down from the machine by less than 1 dB (actually about 0.6 dB). We interpreted this as strong evidence that the operator could act as an optimum non-linear filter. In this experiment we did not expect the optimum non-linear filter (the operator) to outperform the optimum linear filter. We felt that since the RWG noise had no distinguishable features the operator's ability to process the image non-linearly would not give him any advantage over the optimum linear filter. We were pleasantly surprised by the experimental results.

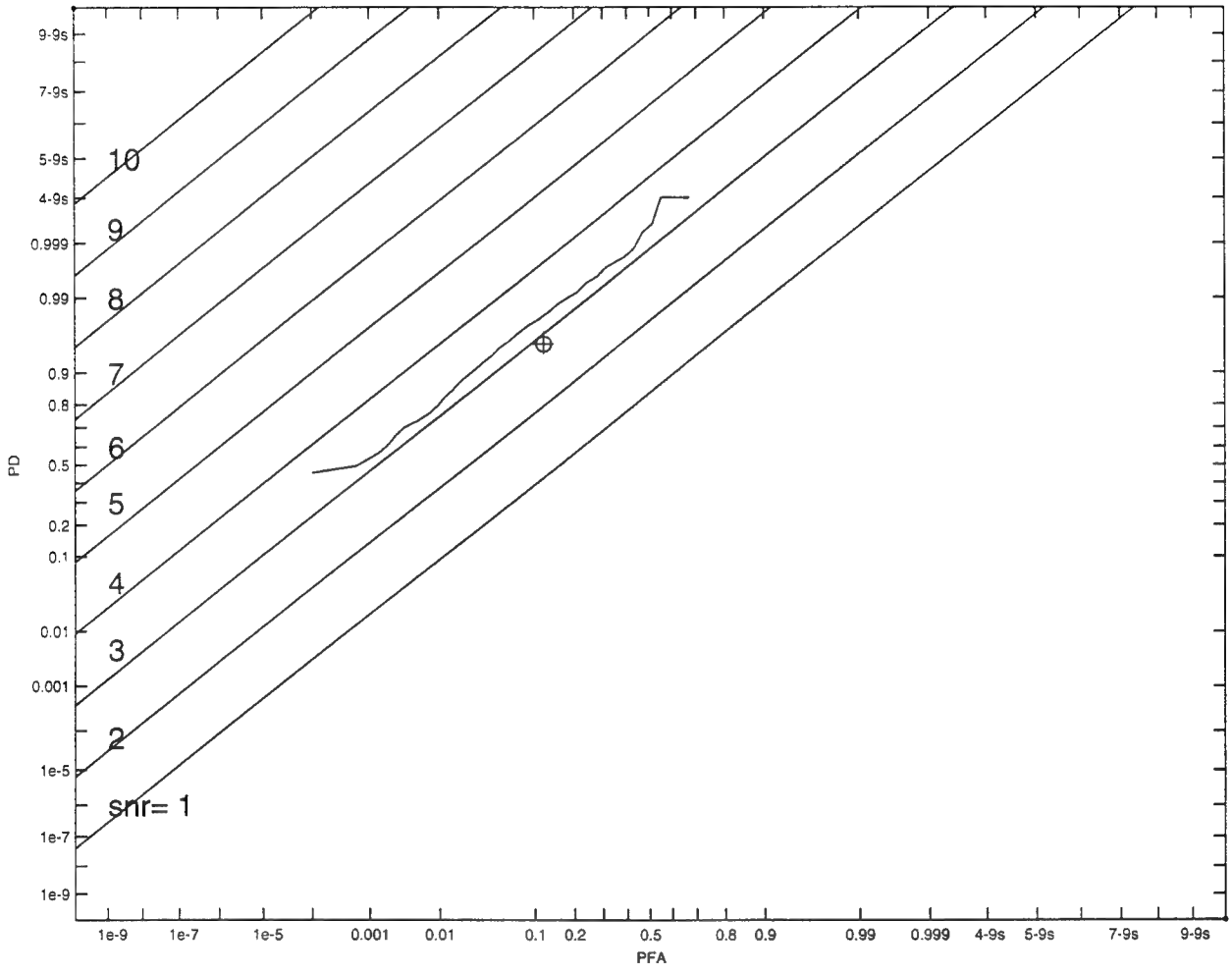


FIGURE 17. Experiment 1. The above PRBPRB graph shows the results of the Experiment 1. Note that the machine performs at the theoretical effective signal-to-noise ratio of 3.0. The operator's performance is very close to this value, lending credibility to the assertion that the human eye/brain is an optimum (non-linear) filter.

B. EXPERIMENT 2: TRANSPARENT TARGETS IN SPOT BACKGROUND

This experiment utilized Transparent targets in non-Gaussian random noise. We used SPOT images of Jacksonville, Fla. for the background/noise. We loaded these images as gray-scale representations which we displayed on the monitor. An example of the monitor display with these target/backgrounds is displayed in Appendix D. We varied from where in the scene we chose the portion of the background that was used for each detection trial so that operators didn't become familiar with what was presented. The experiment obtained (P_D,P_{FA})-pairs for the operator and for the machine. Two different machine filters were utilized. Filter 1 was the optimum linear filter and filter 2 was a matched filter. Because the background pattern/noise had a PSD that was not constant the matched filter is different from the optimum filter. Results for Experiment 2 are shown for a small diameter (D=1 pixel, a point target) in Fig. 19 and a large diameter (D=21 pixels) target Fig. 18. This experiment produced very interesting results. We found that a human outperformed the machine by a substantial amount only if the machine used the matched filter. However, we also found that the machine equipped with the optimum linear filter consistently outperformed the subject except when working with the smallest targets. These results have significant impact. They indicate that a machine equipped with an appropriate filter can detect targets better than a human when the target signature is well known. Automatic target recognizers may eventually outperform humans under the right circumstances. We want to emphasize that the results of this experiment contradicted our hypothesis. We found no indication that there was any benefit to non-linear processing from these results. Although our hypothesis seemed to be incorrect, we saw the results as positive and potentially useful. The results clearly pointed to the merits of optimum linear filtering.

We were intrigued by the results of this experiment and decided to analyze the optimum linear filter in an effort to understand and explain it's success. We first examined the PSD. Fig. 20 is a logarithmic plot of PSD (vertical axis) versus spatial frequency (horizontal axis). We noted that PSD fell off rapidly with increasing spatial frequency. You may remember that the optimum linear filter divides the Fourier transform of the matched filter by the PSD. The fact that PSD is much larger at lower frequencies tends to deemphasize the low frequency content of the matched filter and weight more heavily the high frequency portion. This effect is something like differentiation. Dividing by frequency in the frequency domain is the same as differentiating in the spatial domain. This tends to emphasize the region in the vicinity of the target edges. The Army Research Lab and others (see Appendix A) are currently attempting to field an ATR using a form of edge detection.

We believe that work is in a sense similar but not the same as optimum linear filtering. We think that work may benefit from a closer look at optimum linear filtering rather than thinking directly in terms of edge detection.

We further examined the edge characteristics of the optimum linear filter by examining it's pixel values (which may be thought of as filter weights). Fig. 21 is a three dimensional plot of the optimum linear filter for a matched filter with a diameter of 21 pixels. Examine the figure and you will note that the pixel value maxima are associated with the edge region. To show this more clearly Fig. 22 shows the pixel values associated with the diagonal of the spatial frequency plane. It also shows spikes for the pixel values corresponding to the filter edge region.

The results of this experiment indicate that optimum linear filtering may have strong correlation to edge detection. We recommend that researchers doing edge detection work take a closer look at the theoretical basis for this type of detection. We think optimum linear filtering is a good place to start.

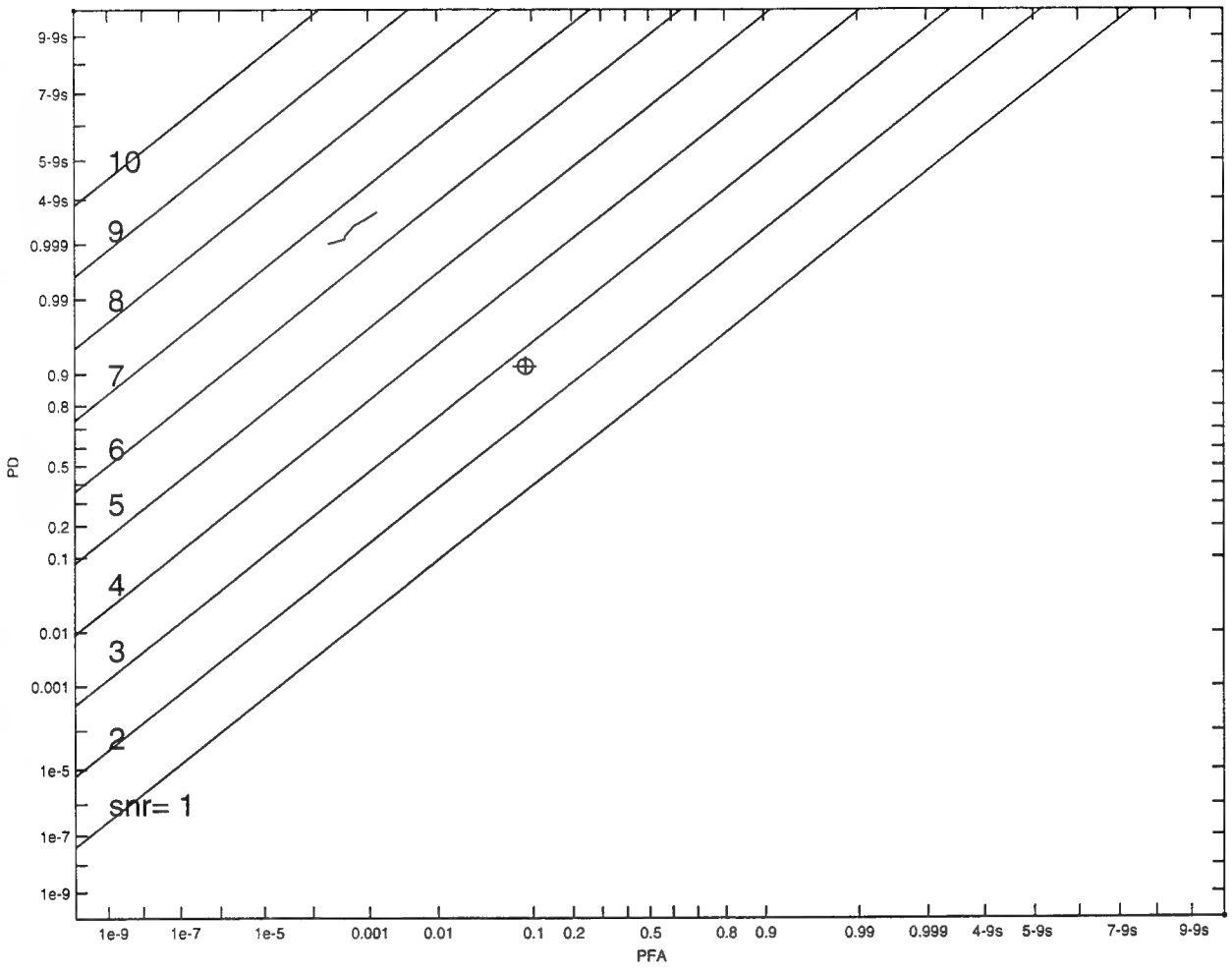


FIGURE 18. Experiment 2, Diameter = 21. The above graph shows the results of Experiment 2. The results were obtained using a target whose diameter was 21 pixels (a large target). Note that the machine when equipped with the optimum filter significantly outperforms the operator at detecting these relatively large targets.

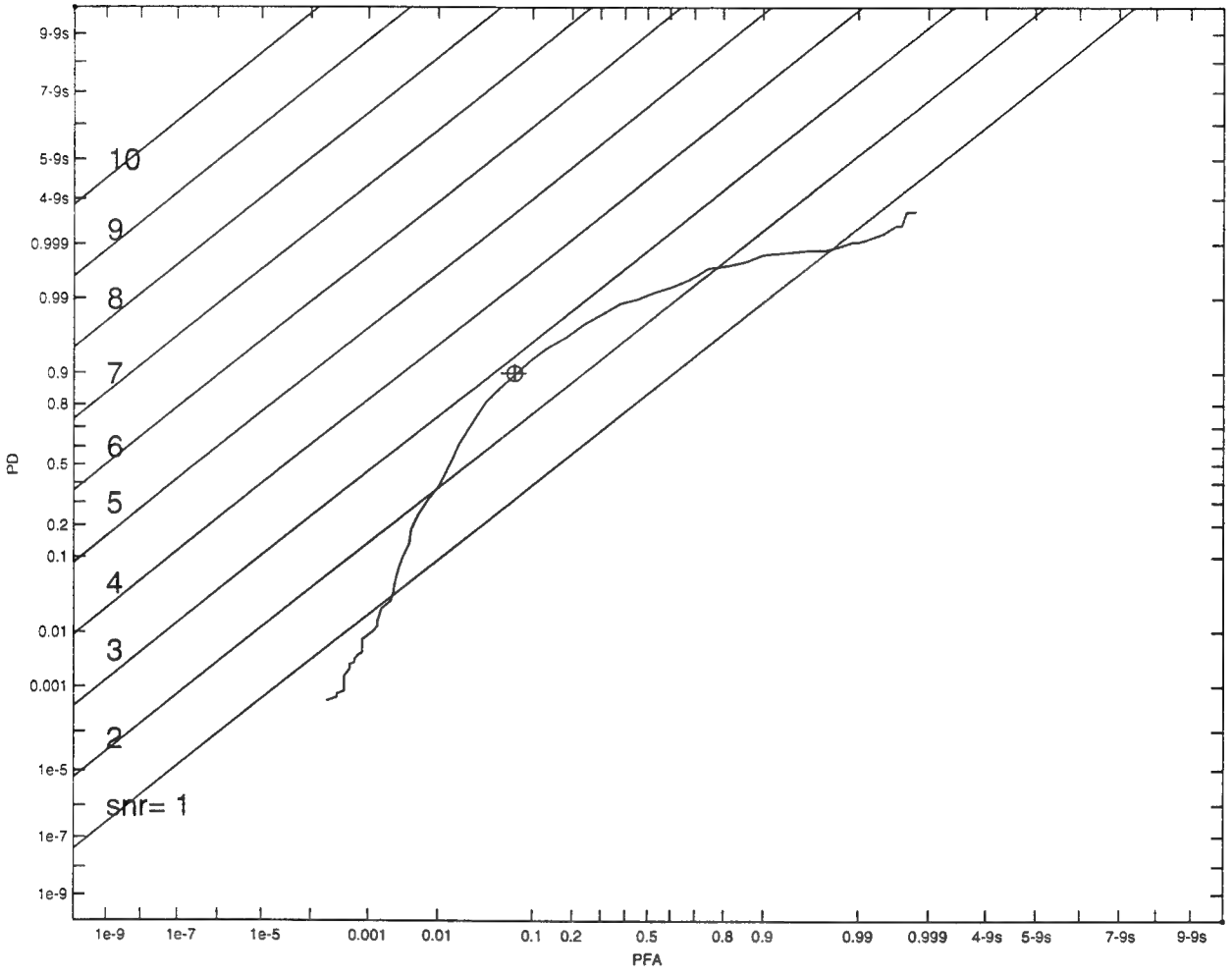


FIGURE 19. Experiment 2, Diameter=1 (A Point Target). Note that the man and the machine/filter achieve equal performance. This is significant because it illustrates that at best the operator's performance will equal the machine/filter. Operator performance even for a very small target signature did not beat the machine.

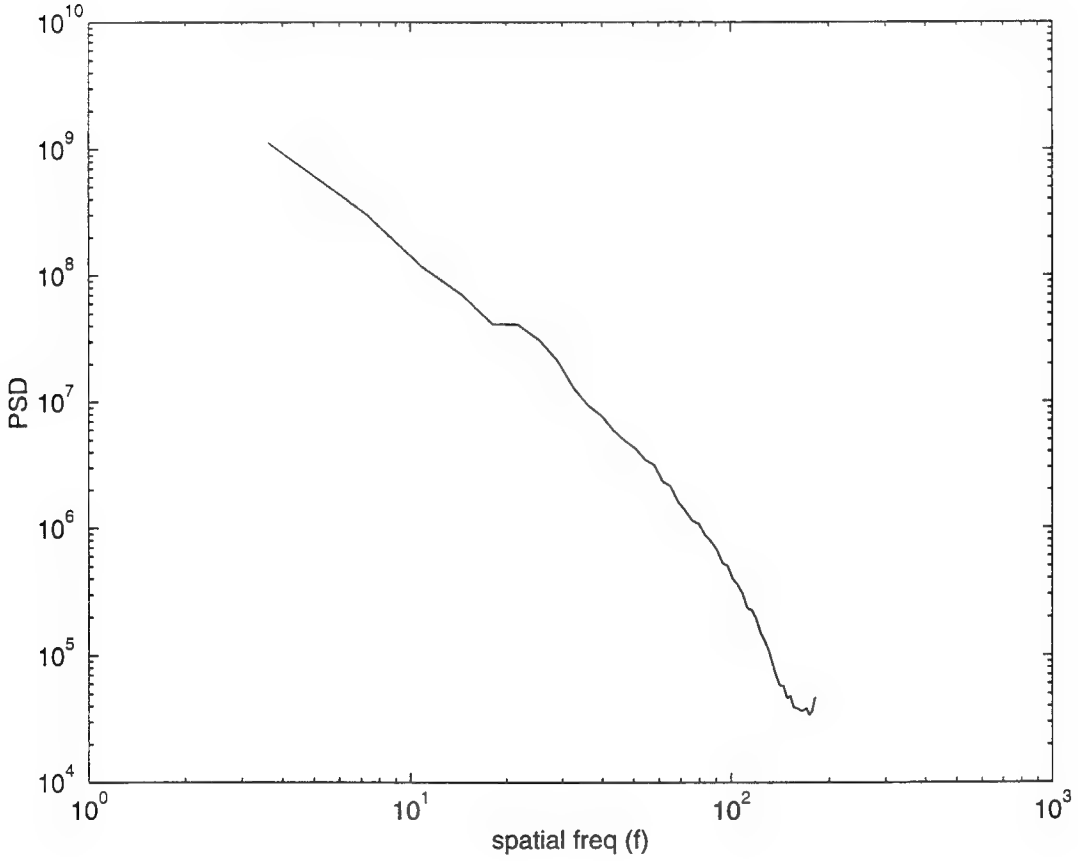


FIGURE 20. Power Spectral Density Plot. The above graph is a logarithmic plot of the SPOT background's power spectral density (vertical axis) versus spatial frequency (horizontal axis). Note that the power spectral density decreases rapidly with increasing spatial frequency.

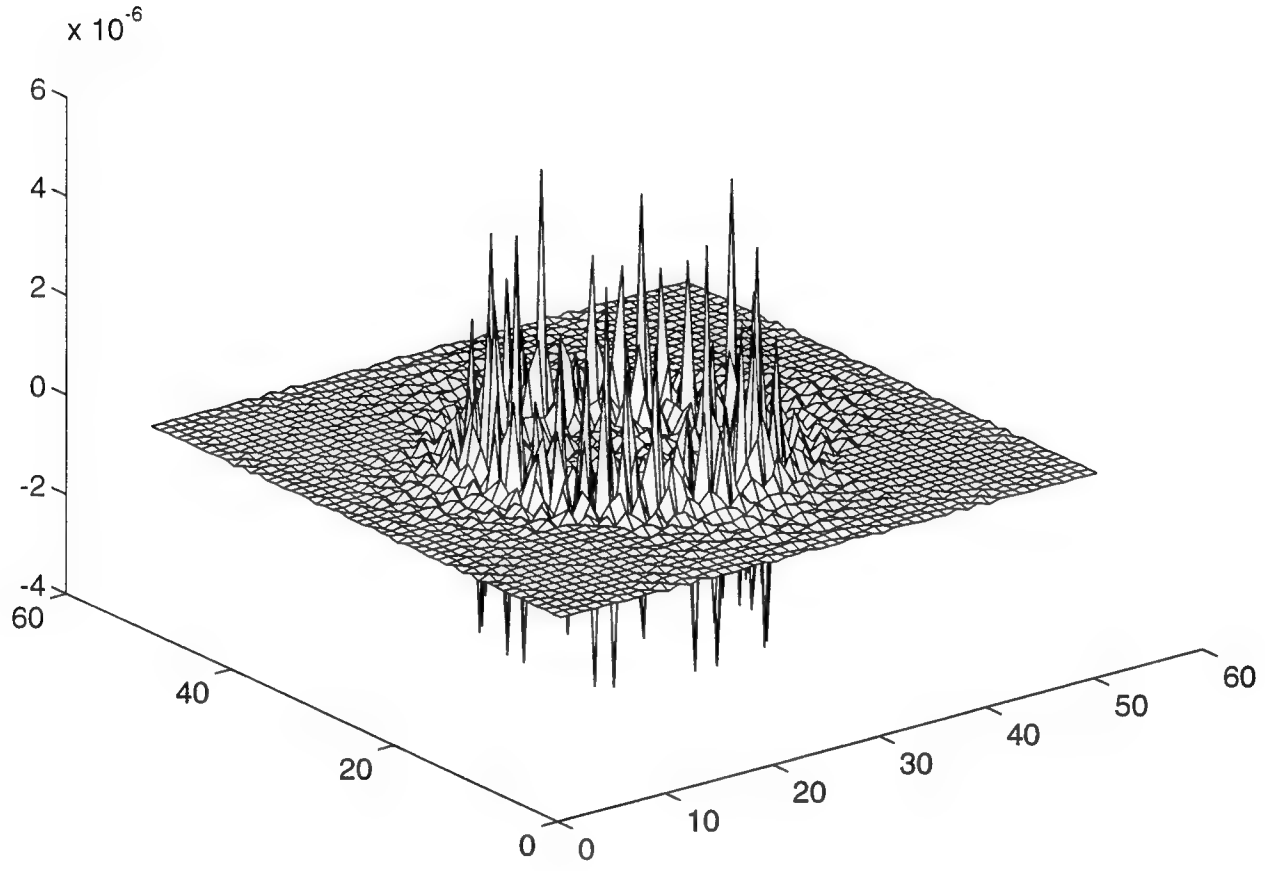


FIGURE 21. Mesh Plot Of Optimum Linear Filter. The above figure shows a three dimensional plot of the optimum linear filter. The spikes correspond to pixel values and the plane parallel with the floor corresponds to the spatial frequency axes. Note that the pixel values are at their maximum values from pixels 15 thru 20 then again from pixels 35 thru 40. This has the effect of emphasizing (applying a greater weight to) the pixels at the target edge region.

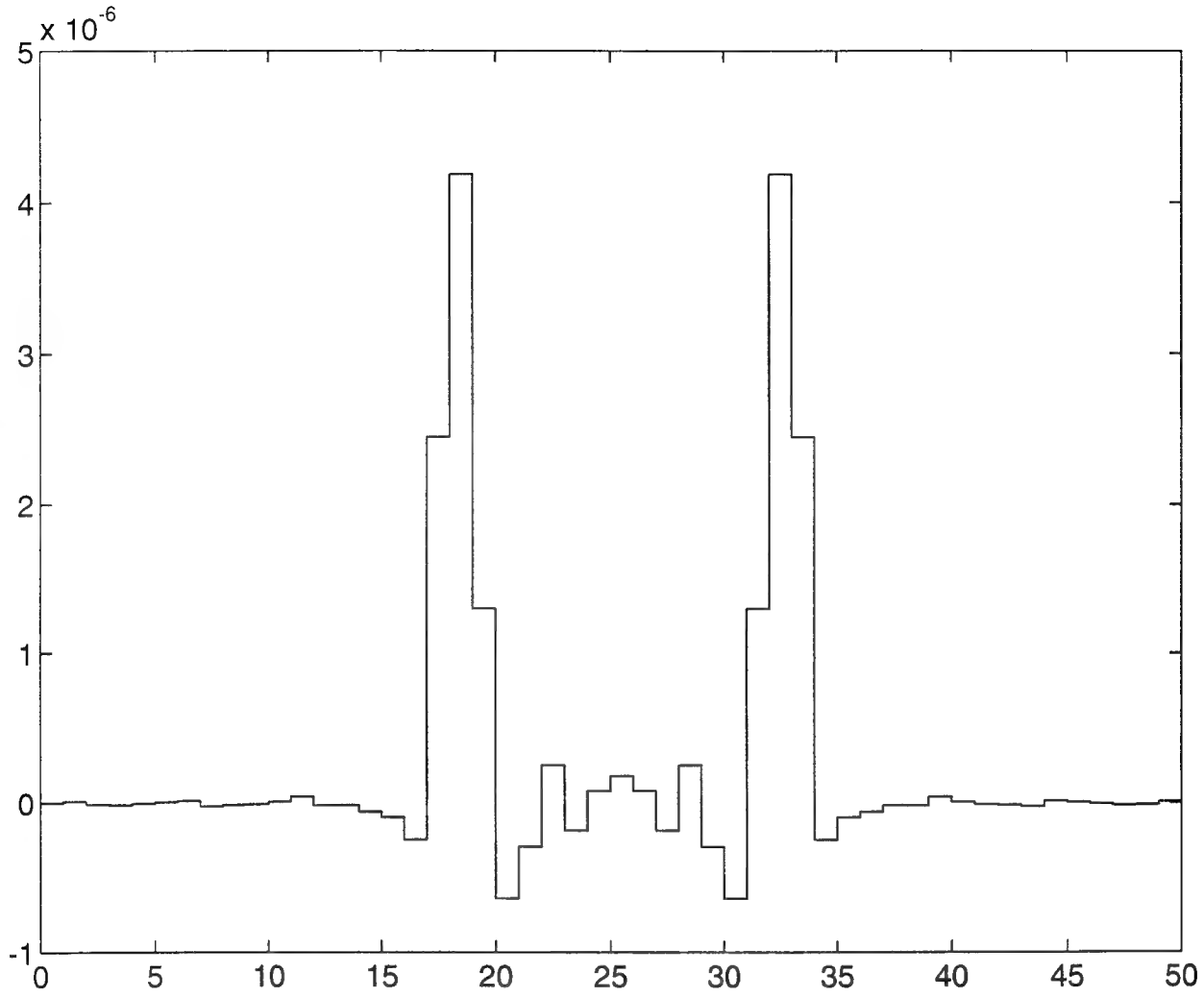


FIGURE 22. Plot Of Pixel Values Along Diagonal. The above figure shows the pixel values(vertical axis) plotted against the diagonal of the spatial frequency plane. The graph allows us to approximate the number of pixels (horizontal axis) associated with some pixel value (vertical axis). Thus we can see that there are approximately six pixels associated with two vertical spikes (maximum filter weights). If you look at the horizontal axis you see that these spikes occur at pixels (17-20) and (31-34). These pixels correspond to the filter edge region. This plot shows that the optimum linear filter applies the greatest filter weight to the filter (and target) edge region.

C. EXPERIMENT 3: CAMOUFLAGED-OPAQUE TARGETS IN SPOT BACKGROUND

This experiment used camouflaged targets in a SPOT image. We found that the operator easily recognized the targets. Even though the camouflage pattern on the target was statistically equivalent to the nearby background pattern, the mismatch between the features in the camouflage pattern and the adjacent background features, particularly along the target's edge was a dead giveaway. Even with zero target signature the operator was able to reliably detect the presence of the target because of the camouflage/edge mismatch. This is remarkable performance, especially when we note that the machine/filter has a very poor ROC for the same set of background and target definitions and parameters. The results of this experiment for a signal strength of two and of zero are shown in Fig.'s 23 and 24. The operator outperformed the machine in both cases. Shown in Fig. 25 is the Camouflaged-Opaque target,with zero signature, in SPOT background. Note that you can readily identify the target because it looks "different" than the background. This target "looks" exactly the same as the background to the machine/filter because the camouflage has zero bias and the target has zero signal. This is why we think non-linear processors will outperform linear filters when only the statistics of the background and target are known.

These experimental results more closely match the original hypothesis. They provide strong evidence that the operator is using a non-linear filtering process to take note of information which the machine/filter cannot use. The question of what processing the subject uses to recognize the targets deserves further investigation. It was not pursued in this thesis. We think the operator might pay particular attention to (i.e., give principle weighting to) the target and background camouflage match, particularly at the target background boundary. We believe an investigation of the decision process that the operator goes through in recognizing a target will yield a high payoff.

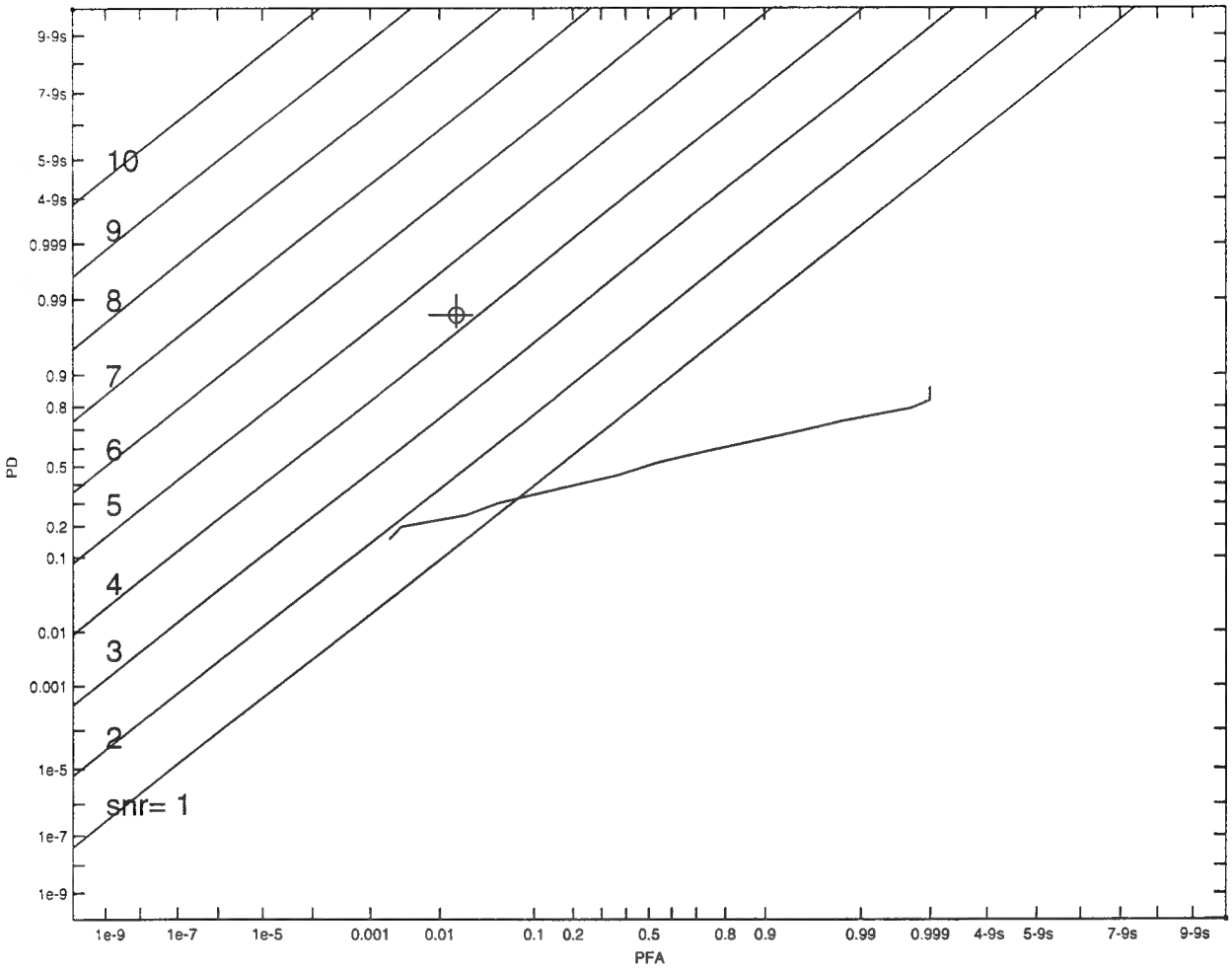


FIGURE 23. Experiment 3, Signal = 1.5. The above figure shows the results of Experiment 3. In the above experiment the target had a signal of 2. The signal strength did not seem to be the most important target characteristic for the operator. The operator seemed to look for the mismatch between the target and the background camouflage. Note that the operator outperforms the machine.

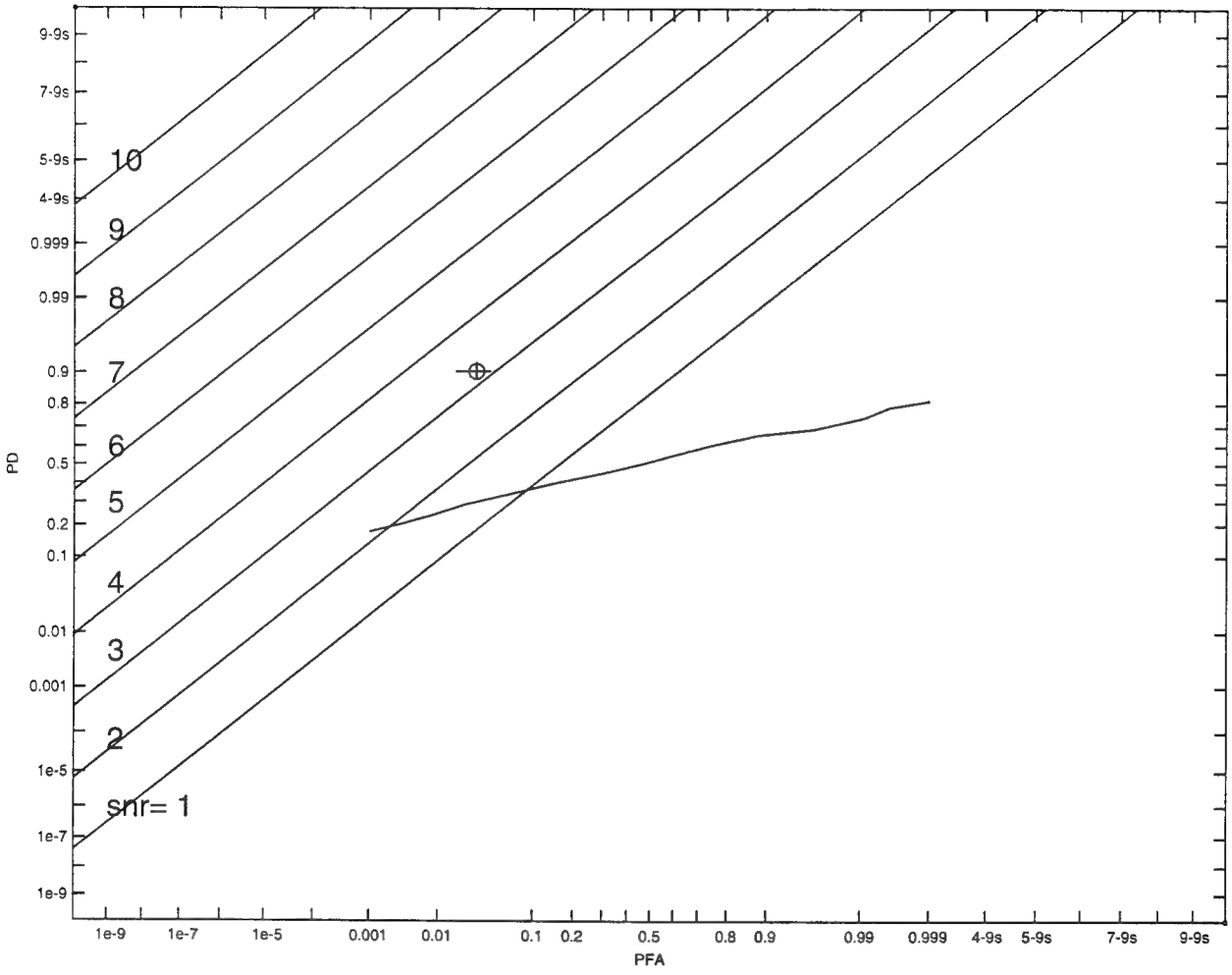


FIGURE 24. Camouflage Opaque Target With Zero Signal. Note that the operator achieves a signal-to-noise ratio effective of 3.1 with zero signal. The operator cannot be using linear filtering because the target and the background have the same signal and therefore look exactly the same to a linear filter.

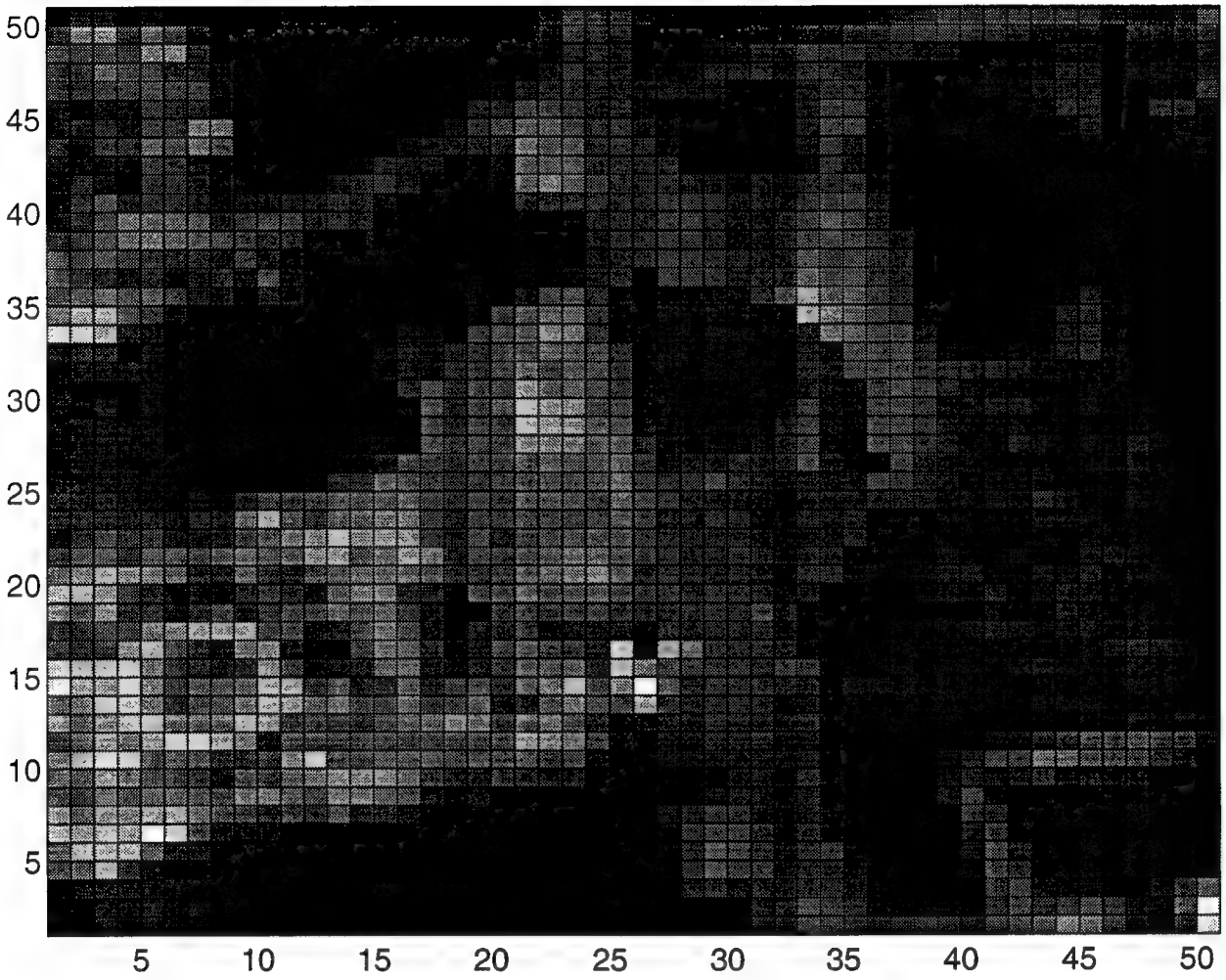


FIGURE 25. Camouflaged-Opaque Target With Zero Signature. The above image shows a Camouflaged-Opaque Target in a SPOT background. This target has zero signature. You can recognize the disc shaped target by the difference in target and background features. You take advantage of the fact that the target "doesn't look right". A linear filter cannot recognize the difference between the target and the background because they both have the same signal level.

V. SIGNIFICANCE

In this chapter we discuss the thesis results and their significance. We believe that the proposed ATR systems which we know of can benefit from our research results and analysis. We make the following observations and recommendations.

We recommend that ATR researchers adopt comparison of the machine/filter device under study with a human operator's performance under the same target/background characteristics as the standard to evaluate how well the machine/filter performed. We found only one other example where this was done in a quantitative sense. This is an important approach and although our research indicates that a human may not be the best possible filter in all situations, a system able to match human performance would have many customers. An ATR as good as a man operating at his leisure would be of great interest and would represent a significant enhancement to combat capability when fielded.

The methodology we adopted of associating ($P_{\rm D}$,$P_{\rm FA}$) together as a pair and using this pair to calculate an $SNR_{\rm eff}$, and then using this $SNR_{\rm eff}$ as the standard of comparison is unambiguous and provides a powerful comparative tool. The research we have read often fails to associate $P_{\rm D}$ with $P_{\rm FA}$ when reporting system performance. Commonplace statements such as "The system has demonstrated its ability to reliably detect targets" together with the remark that "The false alarm rate needs to be improved" are ambiguous to the point of being meaningless. Any system can reliably detect targets if you're willing to tolerate the associated false alarms. The ATR community should adopt the convention of associating $P_{\rm D}$and $P_{\rm FA}$together as a pair. The methodology of reporting $P_{\rm D}$with $P_{\rm FA}$as a pair then reporting a system's performance as the Gaussian equivalent signal-to-noise ratio ($SNR_{\rm eff}$) of that ($P_{\rm D}$,$P_{\rm FA}$)-pair may be used to compare the performance of any two systems, or to compare any system with a human. *

We have examined much work in the ATR community which attempts to use edge detection as the underlying process for target recognition. The results of this work appear promising. We believe that the optimum linear filter taking proper account of the background represents the appropriate

* Some work may only lead to a false alarm rate, $\dot{P}_{FA}$, rather than to a false alarm probability. In such cases the rate , $\dot{P}_{FA}$, should be reported (with P_D) and accompanied by a rough estimate of the rate of false alarm "opportunities,"$\dot{\Omega}$. A rough estimate of $P_{FA} = \dot{P}_{FA}/\dot{\Omega}$ can then be used to calculate $SNR_{\rm eff}$. The non-linearity of the calculation of $SNR_{\rm eff}$ is such that even an order of magnitude error in $\dot{\Omega}$ will have only a minor offset on the calculated value of $SNR_{\rm eff}$.

form of an edge detection algorithm. We recommend that those researching ATR with the edge detection technique reexamine the theoretical basis for this work taking advantage of the similarity between optimum linear filtering and edge detection.

Under certain conditions in the case of a well defined target the optimum linear filter outperforms the human. The circumstances are that the target is known in detail and in advance (in our experiments simply that the target was a disc with well defined edges) and the target does nothing to the background scene except add signature to it (i.e., the target does not hide background irregularities). Under these circumstances the human does not seem to be able to collect and to assimilate all the edge information in arriving at his decision. Further research should be conducted to determine the appropriate background/target situations in which the optimum linear filter may be the most appropriate filter.

We believe that the greatest benefit to ATR research will come from developing an understanding of human non-linear processing in target recognition. This research strongly indicates that a non-linear processor has merit when there are details in the target which can only be characterized in the statistical sense. By this we mean that you don't know exactly what the details of a scene are, you only have to know something about the statistics of that scene. This target situation more closely represents the type of background environment in which a fielded ATR would have to operate. We think that further research in the non-linear process will yield a significant payoff.

APPENDIX A. CURRENT ATR RESEARCH

The search for automatic target recognizers (ATR) began three decades ago in the 1960's. Today there are Army, Navy, and Air Force as well as various commercial programs. Each service recognizes the military significance of a viable automatic target recognizer. Desired Army applications are:

1. Target Aquisition for attack helicopters and air defense systems.
2. Target aquisition for tank and anti-armor systems.
3. Target aquisition for observed indirect fire.
4. Minefield detection.

Desired Navy applications are:

1. Ocean surveillance and targeting.
2. Antiship missile recognition.
3. Land attack weapon guidance, control, and target acquisition.
4. Air target recognition.

Desired Air Force applications are:

1. High value deep target recognition.
2. Cruise missile advanced guidance.
3. Autonomous SAR guidance/target recognition.
4. Image analyst work station data reduction on airborne intelligence platforms. As an emerging technology ATR research is wide open to different problem solving approaches. ATR strategies vary widely. Some of these strategies are listed below:

1. Multi-sensor aided targeting. Intended principally for airborne use, this approach attempts to use the significant advantages of high SNR active radars with advanced algorithms and neural networking, generally in conjunction with a FLIR sensor.
2. Target azimuth, range, elevation and intensity. This approach attempts to use target location, and signature intensity information provided by other sources for target recognition.
3. Edge detection and decision trees. This approach works with optical sensor image data. It compares the difference in boundaries between potential targets and their backgrounds. The approach considers the edge as the most important piece of target information.

Regardless of the approach taken, the ATR under study will produce some characteristic number of correct responses (true detections, correct identification of no target present) and incorrect responses (missed detections, and false alarms). The success of a system can be reported by specifying the number of correct responses and the number of incorrect responses of the system in a given target situation. We have noted that the industry suffers from a general failure to quantitatively report both the correct and the incorrect responses of a system together. Perhaps the least ambiguous quantitative measure of a system's performance is the (P_D,P_{FA})-pair. We prefer this technique and recommend that whenever possible researchers report their results as (P_D,P_{FA})-pairs. The problem seems to be most acute for reporting P_{FA}. We acknowledge that the problem with reporting P_{FA}comes from the difficulty in determining "the number of possible"detections. For instance a detection device that has a one meter resolution examining a 1,000 m^2 area will have fewer detection events than a device with ten centimeter resolution. The number of detection for either device will be related to the method that device uses to sample the background data. We recommend that when researchers find it inappropriate to report P_{FA}directly then they should report the false alarm rate for the device using the most unambiguous units of measure possible. We futher suggest that these reseachers should describe the method used in calculating the P_{FA}rate in the body of the research report and describe in their best estimation how that reported false alarm rate may be correlated with P_{FA}. When reporting system results researchers should understand that users are interested in how the system under study compares with other systems using the same target situation.

In considering the measurement of an ATR's performance it becomes useful to compare the ATR in question with a human. We have noted that the overwhelming majority of research work seems to use targets that are easy for a man to detect. The targets used for ATR appear easy enough for a man to recognize in one rapid pass. Dr. Clarence P. Walters has examined this question. * Dr. Walters obtained the algorithms for a number of ongoing ATR projects and implemented these algorithms on a computer. He then tested these algoritms against humans in similar background/target situations. He concludes that ATR systems and human observers detect targets with the same probability. However the ATR systems produce a false alarm rate that is approximately 40 times that of the human observers. He concludes that humans were able to recognize a wider variety of targets because they train quickly and well. He states that the ATR was able to recognize more small low contrast

* Clarence P. Walters, "Human Perception Testing And Automatic Target Recognition: Performance Against A Common Imagery Set," Briefing to The Army Science Board 1994, Paper for distribution to U. S. Government agencies and their contractors, 29 June 1993. Refer requests to AMSEL-RD-NV-VISPD-LET.

targets than the human but that these low contrast detections probably cause the systems' high false alarm rates. Dr Walters recommends that high quality field imagery should be collected to train and test both ATRs and humans presumably with the intent of using human performance as the goal for ATR. We agree with Dr. Walter's proposal. We think that until truly optimal performance standards predicted by generally accepted ATR theories are identified researchers should compare the ATR device under investigation with human performance under similar target and background parameters and definitions to provide some practical standard by which to measure ATR system performance.

Different ATR systems are using different target/background situations. These situations vary in complexity and difficulty from camouflaged vehicles in a treeline (very difficult) to planes on a runway in the open (relatively easy). When reporting ATR results most researchers don't discuss the level of difficulty of the target/background situation. We recommend that the industry establish a generally accepted method of determining the relative difficulty of different target situations. We think that it would be logical to judge this level of difficulty by asking the question: "how well would a man perform using the same target /background definitions and parameters?" We think that researchers should report the performance of their systems with a discussion of the relative level of difficulty of the target detection problem.

This appendix reviews some of the current Automatic Target Recognition (ATR) work. The sources for most of the material reviewed came from the Proceedings Of the IMAGE UNDERSTANDING WORKSHOP 13-16 November 1994 and from brifings presented to the ARMY SCIENCE BOARD's ad hoc study group on ATR in 1994. When possible we focus the review of the work being mentioned to target Probability of Detection (P_D) and Probability of False Alarm (P_{FA}). Ideally we try to report P_D, P_{FA} together as a pair as we have recommended in the thesis body. However since most of the reports of ATR research do not address target detection or false alarm probabilities directly/quantitatively we report system performance given the performance standards reported by the researchers which are most analogous to (P_D,P_{FA})-pairs. Many of the documents describing various ATR system work neglect reporting detection and false alarm probabilities, presumably because they feel correct target classification implies target detection, or because for the system under study, the authors plans to rely on some other system for target detection (cueing). We believe the difficulty of target detection is underestimated. It seems to us that reliable target detection systems should be developed before before focusing resources on target classification.

We have found it useful to group ATR work into two broad approaches. We denote these two approaches as statistical and intuitive.The statistical approaches include any form of detection whose methodology is dependent upon self adjusting software for recognizing targets with the self adjustment being made using essentially a mathematical/analytic approach exploiting "training" data (in the statistical approach it is the software that is "trained"). This statistical approach includes neural arrays , neural networks, feature extraction algorithms, correlational classifiers.

The intuitive approaches attempt target recognition/detection using essentially 'smart', or enlightened ideas in the preparation of the target recognition software. The "training" data is only used in evaluation of the software's performance– to make the programmer aware of performance problems, so that the software can be revised . In the intuitive approach it is the programmer that is "trained." The intuitive approaches include edge detection, morphology matching, and some forms of feature extraction/comparison.

Statististical detection/recognition depends upon extracting characteristics of a target region then comparing these characteristics with values that are expected for specific target types. The target and background have different distributions of a statistic which is used to label each region as one of two classes. A threshold determines which class a measurement a certain background image statistic indicates. The statistical approach may be extended to include many classes and many features. Statistical approaches typically execute detection using multidimensional distribution or decision trees. The multidimensional approaches are more accurate than the decision tree approaches, but the decision tree approaches are less computationally intensive.

We discuss a few representative statistical approaches next.

The US Army Missile Command is conducting work using a passive multisensor suite. * The system developers are considering three supervised pattern recognition approaches. These approaches are dynamic feature/statistical analysis, neural arrays, and neural networks. In executing feature analysis the researchers state that the approach will limit the number of features to be considered by applying the class overlap partitioning scheme. This scheme feeds the image data to a branch-and-bound procedure or to a genetic algorithm which then selects features to be considered. The set of features selected is applied to an appropriate classifier where a piecewise linear computation is

* S. Richard F. Sims and M. Bowles, "Feature Analysis Versus Neural Arrays Versus Neural Networks For Multisensor Target Recognition," Brussels Sensor Fusion Conference 1993.

carried out. In the report reviewed the researchers claim the system attained 100 percent separation between targets and non-targets.

The neural arrays reported used a vector selection process and the kernels algorithm (k-alg) to provide what they call "exemplars" of the object to be recognized. The result reported by the researchers was a condensed target-knowledge data base incorporating (according to their claims) most of the significant target attributes or features. The resulting set of features was then used to create a projector which they said was such that any input vector if projected onto itself would define a target of interest. The performance of the neural array was reported as amplitude of output for tanks, helicopters, and clutter. The report stated that the neural array does a "good job" of representing training data sets but states that, as the training data sets grows, then so must the computational requirements.

The neural networks reported was the multi layer perceptron using a new self partitioning neural network (SPNN) methodology. The image was feature mapped using a frequency transformation prior to being fed into the neural net. This network used 40 input nodes, 20 hidden layer nodes and one output layer node for each target or target class. Partitioning is done in one pass while each training image vector is applied to the network. The researchers state that no training or updating of the network weights occurs during the conflict evaluation pass. The errors from each node are propagated with the typical back propagation and stored as an error vector. Each set of error vectors are then compared to form a conflict matrix. A conflict threshold is used to determine a clustering of the training vectors. The researchers state that each cluster is used to train individual networks identical to the original one used to generate the error vectors. The researchers report that the partitioning approach requires fewer passes to train the net than other neural net approaches. However, the designers can not predict the number of networks that can be generated and thus cannot predict computational resources needed in different situations. The researchers reported that the neural network tested achieved 100 percent recognition in over 200 iterations.

The direction the Army Missile Command seems to want to take in the future is to combine the above algorithmic approaches in the same system and use it in conjunction with a human operator in a supervisory role to identify close range high signature targets.

The Davis Sarnoff Research Center is investigating combining neural networks with pyramid

representations for target recognition.* The researchers claim that the relatively simple image features produced using their "pyramid" image reduction algorithm are more effective neural network inputs than more complex object-tuned features. The system conducts a coarse-to-fine search, and is capable of context learning and data fusion. The most succesful experimental results seem to come from a system that employs a hierarchical network capable of employing what are termed "road maps." As an example, generation of a road map from an image may be thought of as the assignment of pixel weights to an image by measuring the distance of a region of interest from a fixed relatively long linear feature. The Sarnoff study compared the pyramid technique with a similar technique using eigenfeatures. They reported that the pyramid technique appears to be more robust. The initial results of this research in detecting buildings in a rural setting (a relatively easy target) were reported as a ROC curve with true positives (as a percent) reported against false positives (as a percent). The (P_D,P_{FA})-pairs thus reported ranged from (0.85,0.03) to (0.94,0.12).

Neural Networks Laboratory and the Illinois Institute of Technology are also researching a neural network approach to target detection.* The neural system consists of a shape description network, a geometric normalization network, and a recognition stage using what they call fuzzy pyramidal neural networks. This research uses two innovative approaches. First is a novel pyramidal architecture called "the vectorial gradual lattice pyramid" which is a vectorial space representation of shape. Second is a method to determine feature tokens called "cancellation energy." The cancellation energy approach uses directional information in the vectorial space representation of shape. The targets used in the research appear to have been aircraft in the open. The results of the research reported were that the recognition rate was between thirty five and one hundred percent depending on the target orientation and scaling factor used. The research also reports that "invariant recognition with similarity transforms as well as moderate affine transforms are possible ." It is noteable that the research does not mention measurements of false alarm rate.

The Hughes Aircraft Company is researching the what they call Self Adaptive Hierarchical Target Identification and Recognition System (SAHTIRN).† SAHTIRN applies a fast prescreening

---

* Clay Spence *et al.*, "Neural Network/Pyramid Architectures That Learn Target Context," Proceedings of Image Understanding Workshop, Vol. II, pp. 853, November 1994.

* Jezekiel Ben-Arie, K. Raghunath Rao and Zhiquian Wang "A Neural Network Approach for Shape Description and Invariant Recognition," Proceedings of Image Understanding Workshop, Vol. II, pp. 863, November 1994.

† Cindy E. Daniell *et al.*,"Artificial Neural Networks for Automatic Target Recognition," Proceedings of Image Understand-

process over an entire field of regard to select potential targets for further processing. SAHTIRN then classifies potential targets as targets or not targets (*i.e.*the system does target detection). For those identified as targets SAHTIRN specifies the type of target it is (*i.e.*the system does target classification). The researchers claim that SAHTIRN accomplishes this using three stages of neural networks. These stages are:

1. Segmentation (forms coherent boundaries out of many individual local edge elements):-
2. Feature extraction and pattern coding using what is refered to as a "neocognitron":- (A hierarchical feed-forward system is organized into several homogeneous layers. Transformation between layers is accomplished by convolutional masks, which serve as connections between neurons.)
3. Classification of the specific target through the use of a back propagation learning rule:- (This back propagation procedure is implemented in a parallel computer that, in general, having been shown the appropriate input/output exemplars specifying some mapping function, programs itself to compute that function.)

The study reported successful ATR performance when looking for targets on a terrain board. The test results seemed to focus on correct target classification not detection. The reported results include probabilities of correct and missed classifications and probabilities of rejection. Results for the detection stage are not reported in a form that allows determination of the values for (P_D,P_{FA})-pairs.

HNC Software INC is researching many recognition approaches at least two of which are statistical and merit mention here. The two approaches execute target recognition through "context vectors" * or the k nearest neighbors.† The context vector approach is described as using affine wavelet transformations to define a set of image primitives. These primitives are treated as comprising a vocabulary of image features. The system is designed to provide content address ability of images based on similarity of text or images. The system highlights portions of retrieved images that most closely correspond to user queries then assigns index terms to unlabelled images based on similarity of content to a trained set of images. No (P_D,P_{FA}) performance results were provided for

---

ing Workshop, Vol. II, pp. 853, November 1994.

* Robert Hecht-Nielson *et al.*, "Fast k-NN Search for Robust ATR Object Matching," Proceedings of Image Understanding Workshop, Vol.II,pp. 889

† William R. Caid, "A Context Vector Approach to Image Indexing and Retrieval," Proceedings of Image Understanding Workshop, Vol. II, pp. 885, November 1994.

the context vector approach. The application is perhaps unique in that it is stated it has commercial applications. The system is currently being studied for use in text based retrieval systems such as an automated library retrieval system.

The k nearest neighbor approach is designed to recognize partially obstructed targets. In this technique targets are first defined by spatial frequency based feature vectors. The obstructed target feature vectors are then matched against a large feature vector reference data base to find reference feature vectors with at least fifty percent matched features. The matching algorithm uses a generalization of Euclidian distance called Rousseeuw distance. The system finds the k nearest Rousseeuw distance matches using a voting operation. This voting operation decides the class membership of the target. The system requires very large target databases and does not address target detection. No (P_D,P_{FA}) results were reported for the k nearest neighbor approach.

The US Army research Laboratory is investigating relational template matching (RTM) for target recognition/detection.* RTM uses a model based multisensor ATR algorithm to recognize military targets under a variety of different environmental conditions with ranges to targets of 1-6 kilometers. The researchers claim the ATR system applies an optimization methodology and an ATR algorithm. The optimization process (done once for a given set of targets) partitions and categorizes target signatures. This optimization process has a high degree of computational complexity. The system uses the relational templates to discriminate between target categories. A recognition algoritm compares aspects of a target with known aspects of different classes of targets. These target aspects are found by application of the edge detection technique. The edge detection technique uses many sets of probes (each probe being two points) we believe taking advantage of the theory that the intensity change across a target boundary is on average greater than in the local background. The many values found by the target probes determine the target aspects which are compared with the known aspects of different targets. The work reviewed did not reveal the results of this research in terms of (P_D,P_{FA})-pairs.

Discussed next are some intuitive approaches to target recognition/detection.

Lincoln Laboratory and MIT are researching what they call The Experimental Target Recog-

---

* Ms Teresa Kipp, "ATR Relational Template Matching," Briefing to The Army Science Board 1994.

nition System (XTRS).[†]Researchers claim XTRS detects and recognizes armored tactical vehicles in registered range and intensity images provided by ground based and airborne laser radars. They say that XTRS can recognize very noisy images at arbitrary distances and orientations. The key innovation is a generic matching engine that compares image events (i.e. silhouettes) in the form of constituent primitives to some appearance model (AM) hierarchy. XTRS consists of four subsystems:

1. Event characterization describes the relevant features (image events) of the input imagery. Events are transformed into "range silhouettes" using "edge detection" and "region segmentation." The report states that the event is decomposed into an "event primitive" which is output to the model library. In edge detection the image event is first cleaned (all isolated dark pixels are removed) then convolved with a difference-of-Gaussian kernel. The so-convolved output is examined for "zero crossings" which correspond to the edges in the image. These zero crossings have characteristic length, average strength. These characteristic aspects form the event primitive. In range segmentation detection the image event is also cleaned then examined for interesting features. Each interesting feature forms an "interest image." The interest image with the most interesting aspects is selected and contour outlined to form an event primitive.
2. The model library contains the targets to be recognized stored as Appearance Models (AM's). AMs are descriptions of expected target appearance over a useful range of aspects. Two different model libraries are used depending on whether the event primitive was generated from edge detection or region segmentation. The event primitive is then matched to the model from the model library.
3. Matching is the process of comparing the extracted event and it's constituent primitives to the models in the library. The same matching engine is used for edge detection primitives and for region segmentation primitives.
4. Control orchestrates the operation of the other three components.

The XTRS system research reported recognition results in terms of correct classification. The report did not specifically address target detection . Depending on target type the system reported a 65- 98 percent classification success rate.

The College of Engineering of the University of California, Riverside is researching a model based

---

† Jacques G. Verly *et al.*,"Model Based System for Automatic Target Recognition from Forward-Looking Laser Radar," Optical Engineering, Vol31 No 12, pp. 2540, Dec 1992.

target recognition system that uses a hierarchical Gabor wavelet representation.* The approach combines the global and local Gabor-based measures for target indexing into a model data base. The ATR system uses the Gabor wavelet to decompose target models and image data and a Gabor grid (which can be thought of as a topology-preserving map) to efficiently encode both signal energy and structural information of a target in a sparse multi-resolution representation. The system uses a flexible matching mechanism based on the Gabor decomposition and the Gabor Grid. Gabor frequency is used to estimate the scale variation of a target from a model. Gabor magnitude is used for probe matching based on local structural energy patterns. Gabor phase is used to evaluate the matching result in terms of average local image displacement between the model and the data. The researchers report that the system does not rely on shape features or contours for target detection and reports being tolerant of distortions of viewing scale, aspect, and environmentally induced deviations. The reported results are that a 98 percent recognition rate is achieved in 207 recognition experiments. The research further reports tolerating signature variations of 20 degrees in depression angles of 22.5 degrees with up to fifty percent occlusion. No (P_{D},P_{FA}) results are obtainable from the report.

The Beckman Institute for Advanced Science and Technology is researching a two dimensional edge detection scheme for general visual processing.* The researchers report that the scheme constructs a 2D edge detection functional to detect edges under the guidance of the Laplacian of Gaussian zero crossing contours. They claim the detection functional is optimal in terms of SNR and edge localization accuracy for detecting edges in 2D images. The researchers claim the detection technique provides:

1. An edge regularization procedure that enhances the continuity and smoothness of the detected edges except at corners.
2. An adaptive edge thresholding procedure that is based on a robust global noise estimation approach and two other physiologically originated criteria.
3. A scale space procedure that combines edge detection results from different scale channels reliably. The researchers do not report the performance of their system in terms of detected

---

* Xing Wu and Bir Bhanu,"Target Recognition using Multi-scale Gabor Filters," Proceedings of Image Understanding Workshop, Vol.I, pp.505, November 1994.

* Richard J. Qian and Thomas S. Huang,"Optimal Edge Detection in Two-Dimensional Images," Proceedings of Image Understanding Workshop, Vol.II, pp.1581, November 1994.

targets or edges. They offer three images of what looks like a sketch of a human subject with distinct edge characteristics as evidence that the system fine tunes edge detection results to make them similar to what is percieved by human visual systems.

In reviewing the literature related to ATR we tried to compare the results of the different techniques in effort to gain a quantitative assessment of the relative success of one system to other systems. We have noted that this is impossible to do. Much of the problem lies in the relative difference in the types of targets each system is designed to look for and the background clutter the system designers tend to put those targets in. We accept these differences and understand that target recognition performance is situationally dependent. We have also noted a general failure on the part of ATR designers to report the performance of their system in terms of (P_D,P_{FA})-pairs. In particular there is almost a reluctance to report the performance of a system in terms of false alrm rate, and when the false alarm rates are reported they are often sufficiently ambiguous to be misleading. We think this is a self defeating approach that fails to clearly present the significant gains made by the community in the past decade, or to clearly represent the operational capability of the present systems which may have tactical application in a limitted scenario today. We recommend that the industry study the question of how best to standardize the reporting of ATR performance, so as to develop a meaningful way to compare the techniques and systems being studied without destroying the creativity needed in the problem solving process.

APPENDIX B. EXPERIMENTAL DATA

This appendix contains data for Experiments 1,2,3 not included in the main body of the thesis. This data is displayed on probability-probability (PRBPRB) graphs which are discussed in the thesis main body (Chapter I Introduction).

1. EXPERIMENTAL DATA. EXPERIMENT 1

The following pages contain Experimental data for Experiment 1. The data is shown as PRBPRB plots of an operator (P_D,P_{FA})-pair and a matched filter (which in the type of background used in these experiments is the optimum linear filter) ROC.

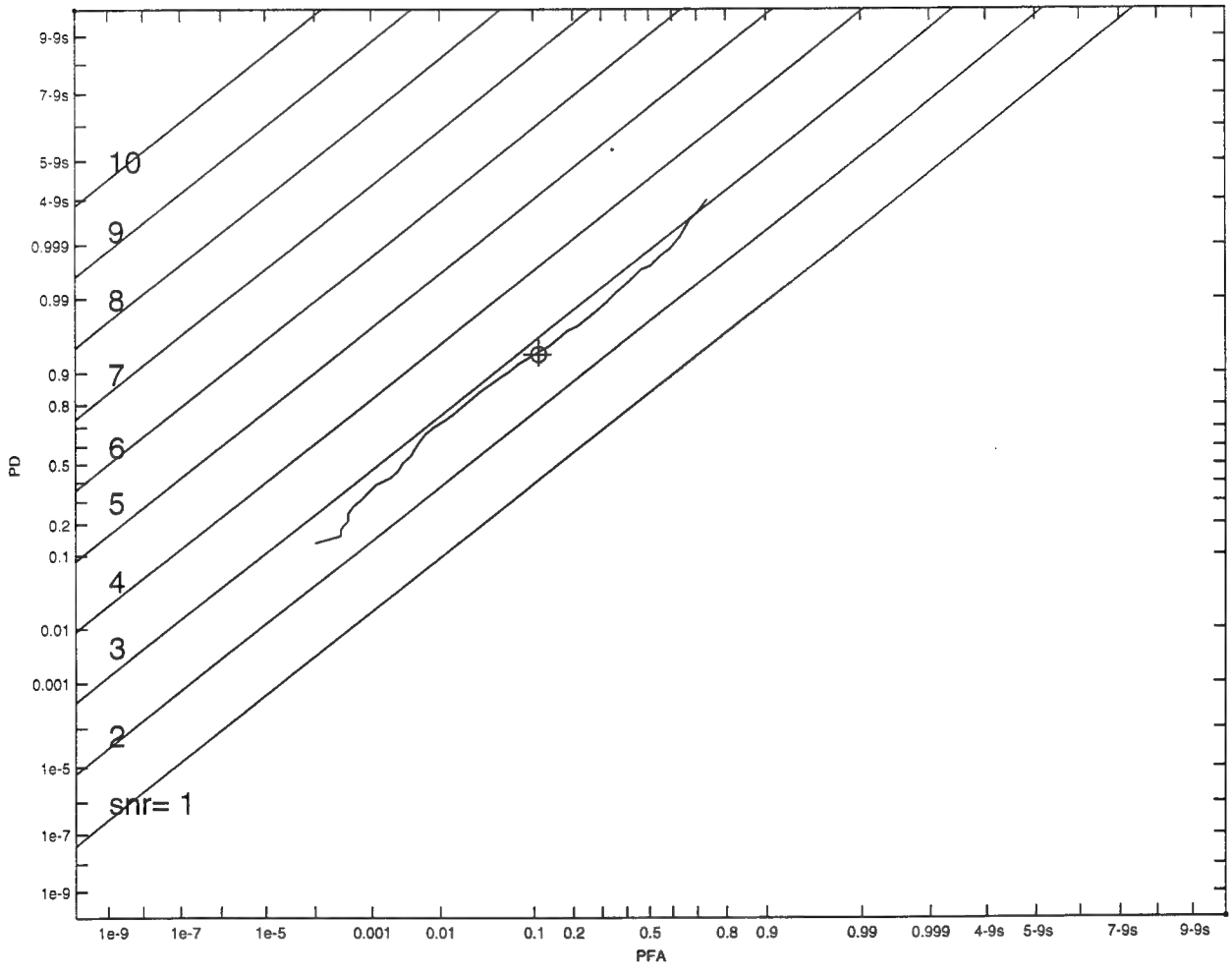


FIGURE 26. Experiment 1 Snrnom=3, Diameter=21

2. EXPERIMENTAL DATA. EXPERIMENT 2: OPTIMUM LINEAR FILTER

The following pages contain experimental data for Experiment 2. The data is shown as PRBPRB plot of an operator (P_D,P_{FA})-pair and an optimum linear filter ROC.

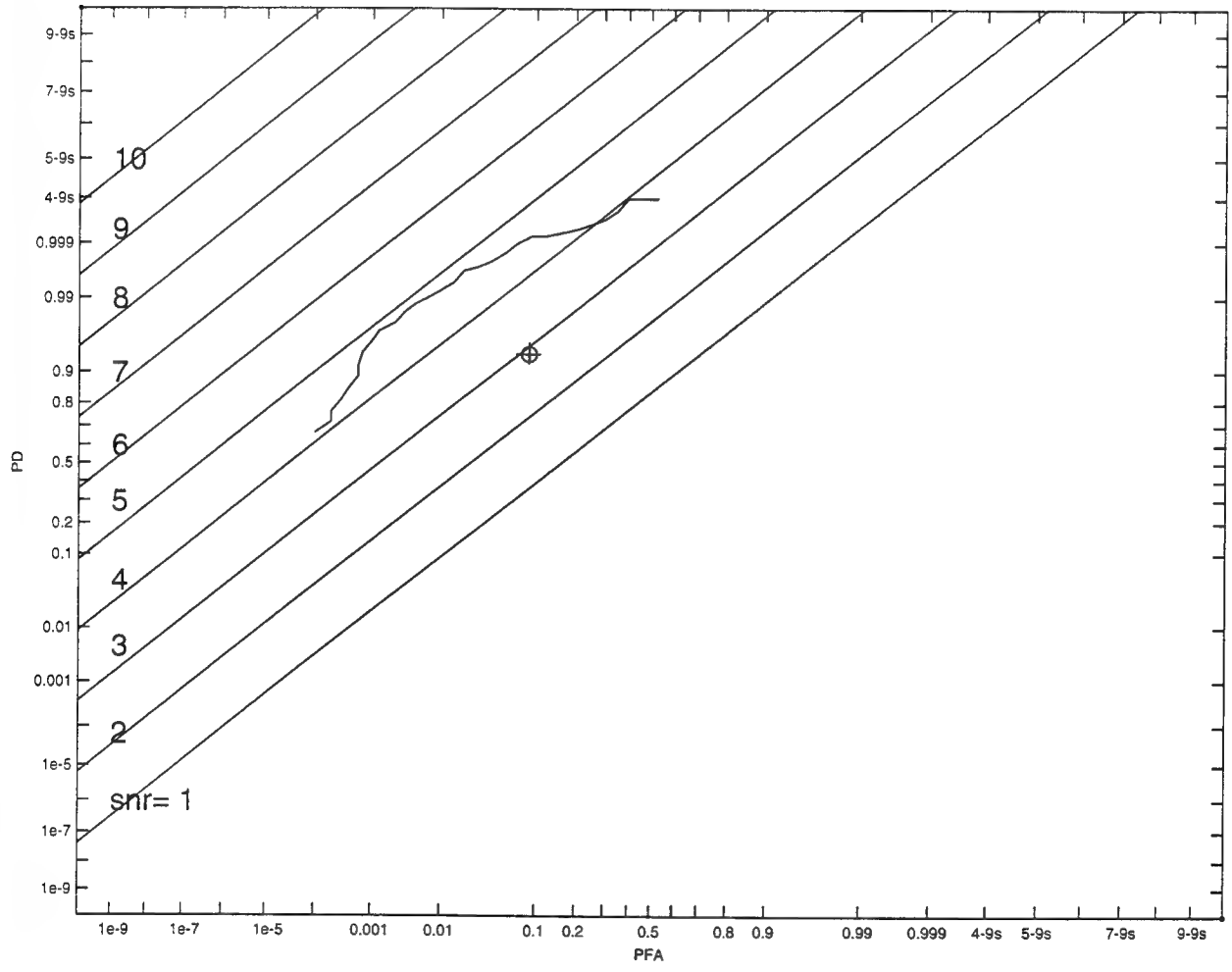


FIGURE 27. Experiment 2 Signal=4.0, Diameter=11.

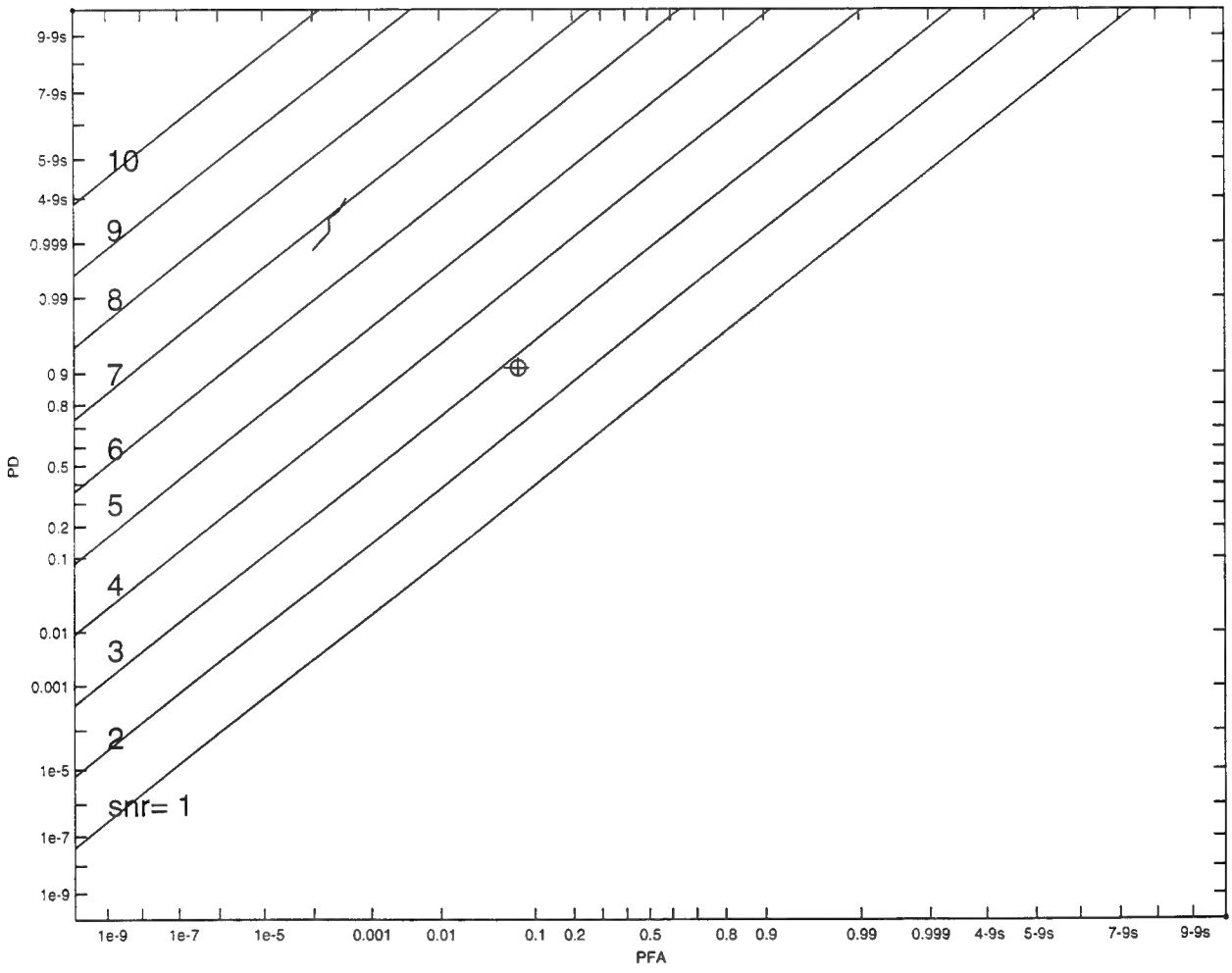


FIGURE 28. Experiment 2, Signal=4.5, Diameter=11.

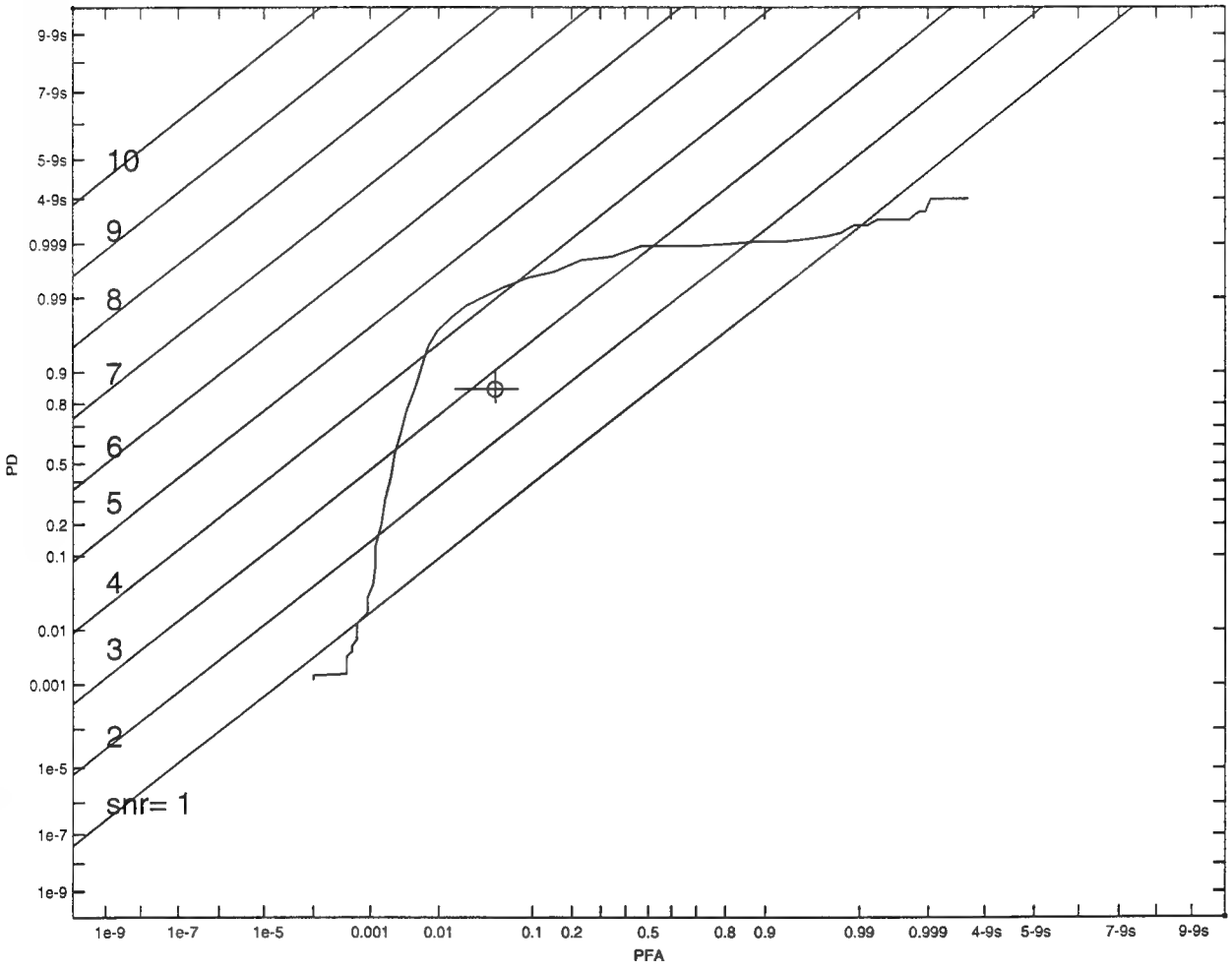


FIGURE 29. Experiment 2 Signal=4.0, Diameter=3.

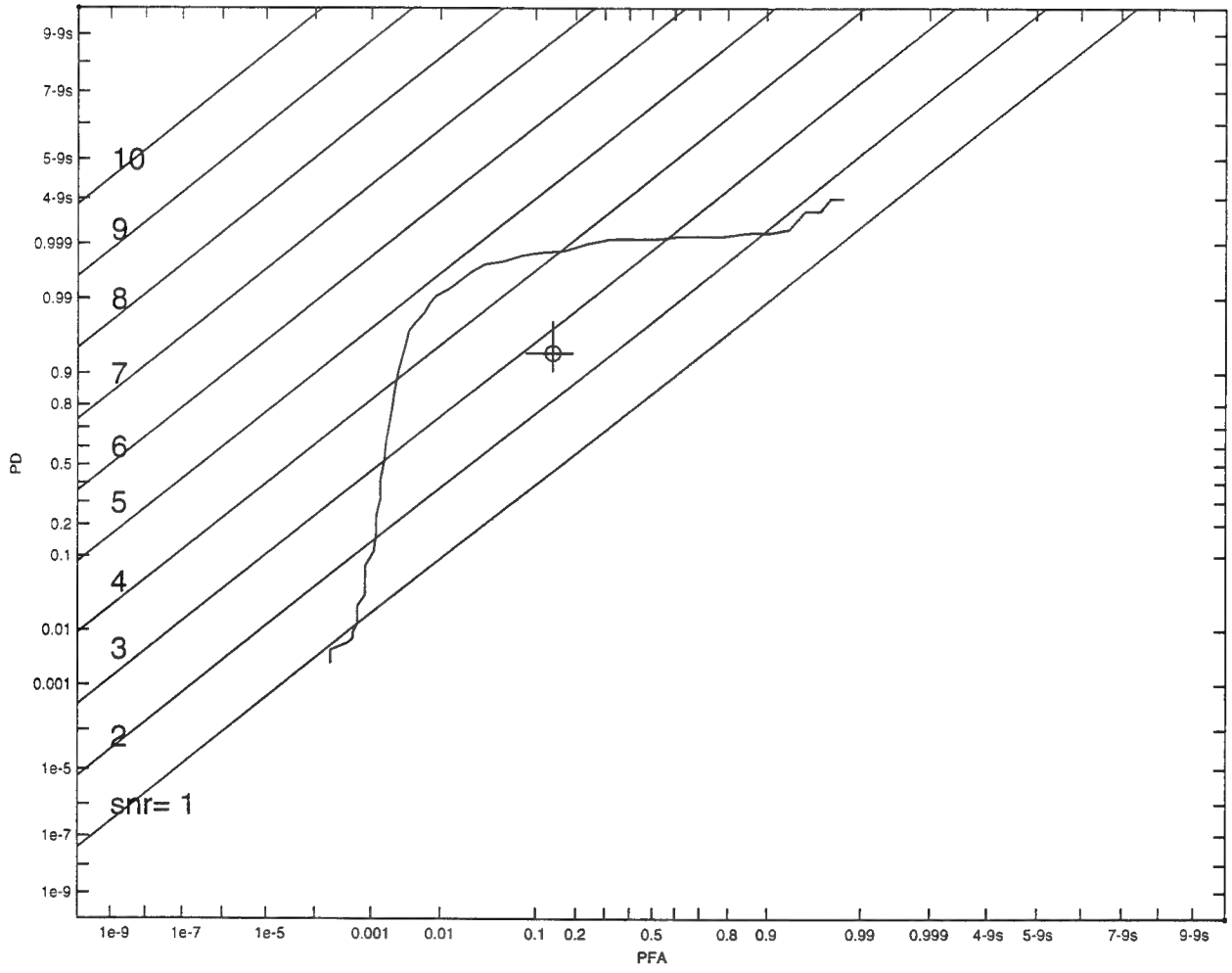


FIGURE 30. Experiment 2 Signal=4.0, Diameter=5.

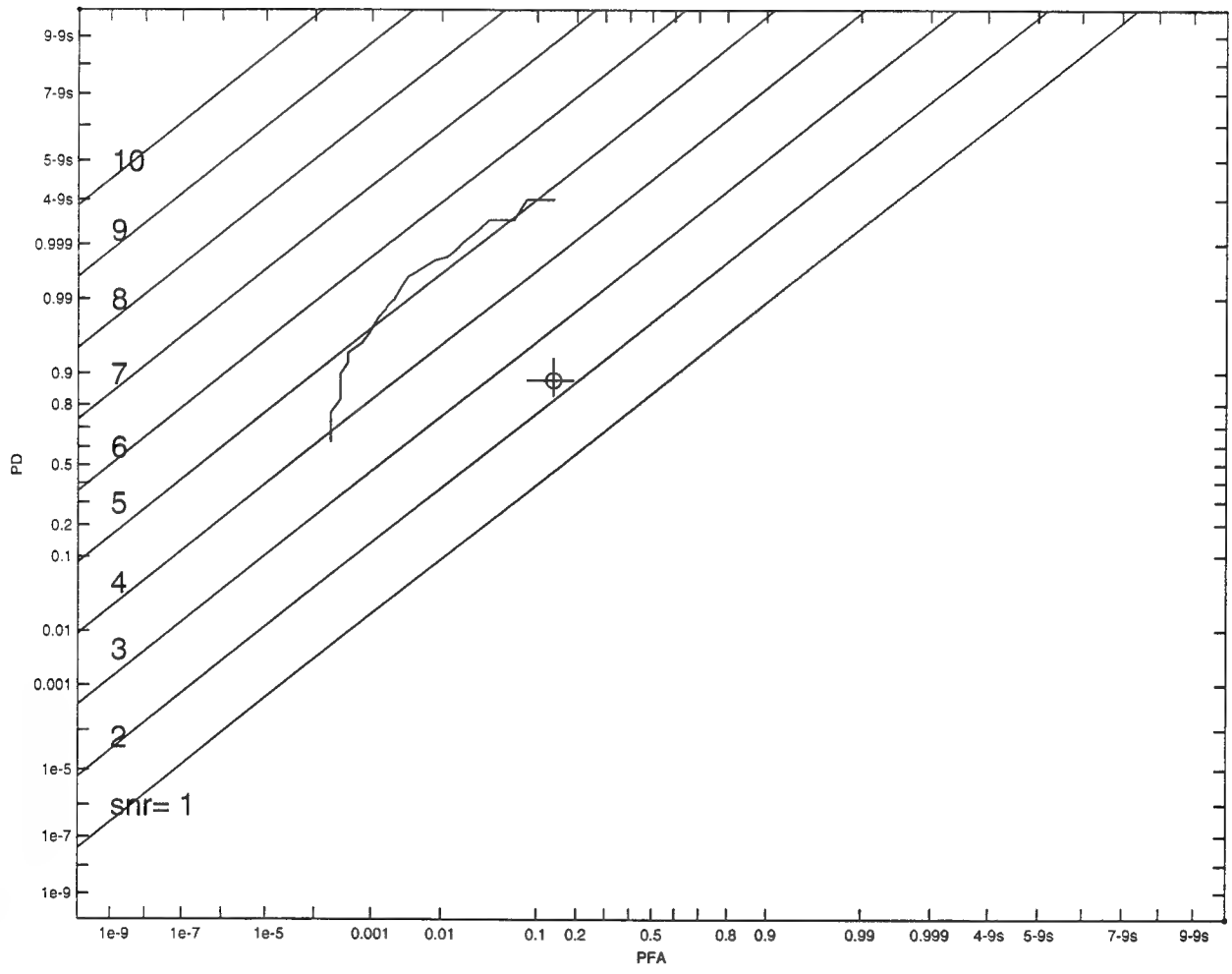


FIGURE 31. Experiment 2, Signal=4.5, Diameter=9.

2. EXPERIMENTAL DATA. EXPERIMENT 2: MATCHED FILTER

The following pages contain experimental data for Experiment 2. The data is shown as PRBPRB plots of an operator (P_D,P_{FA})-pair and a matched filter ROC.

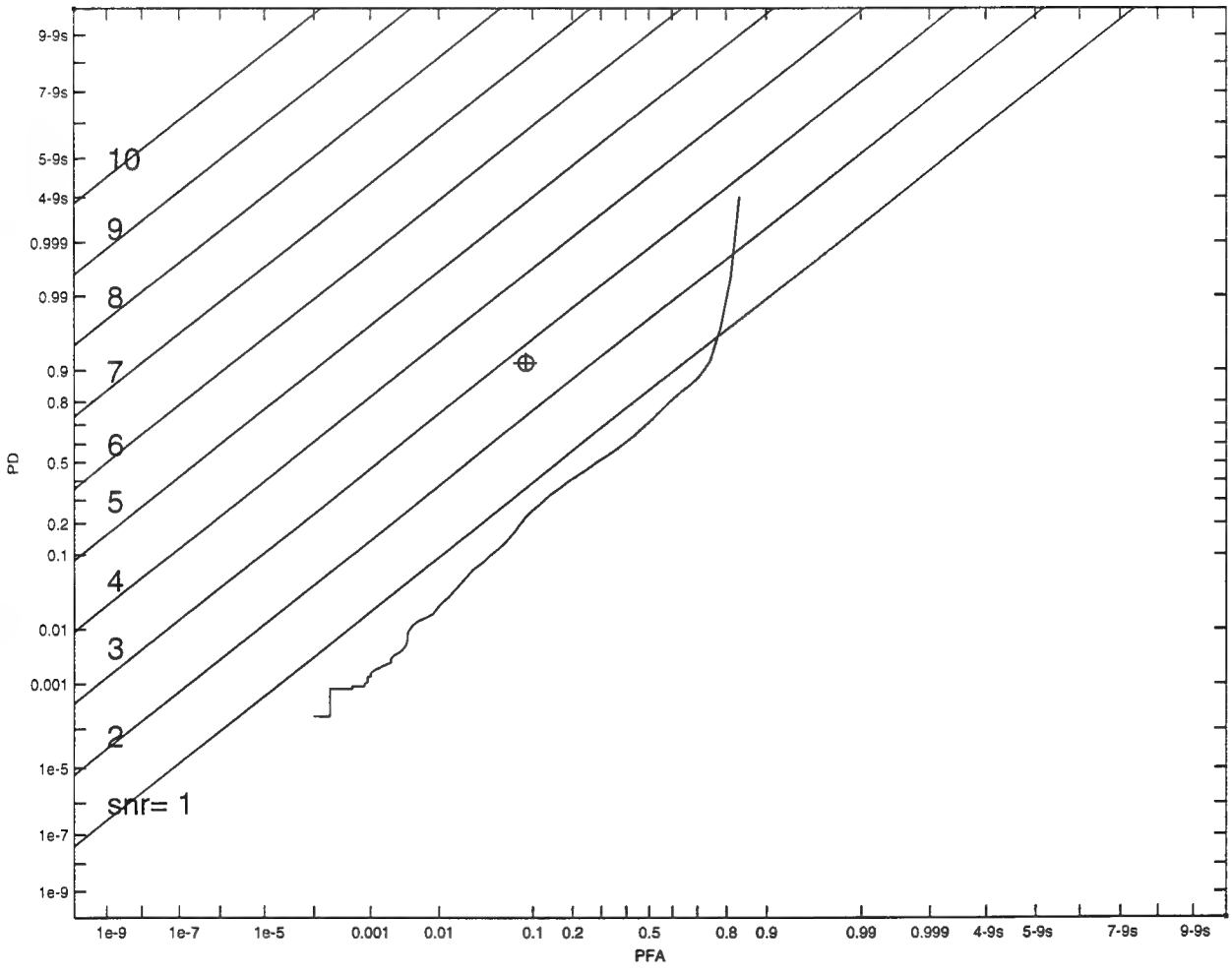


FIGURE 32. Experiment 2 Matched Filter Signal=4.0, Diameter=21.

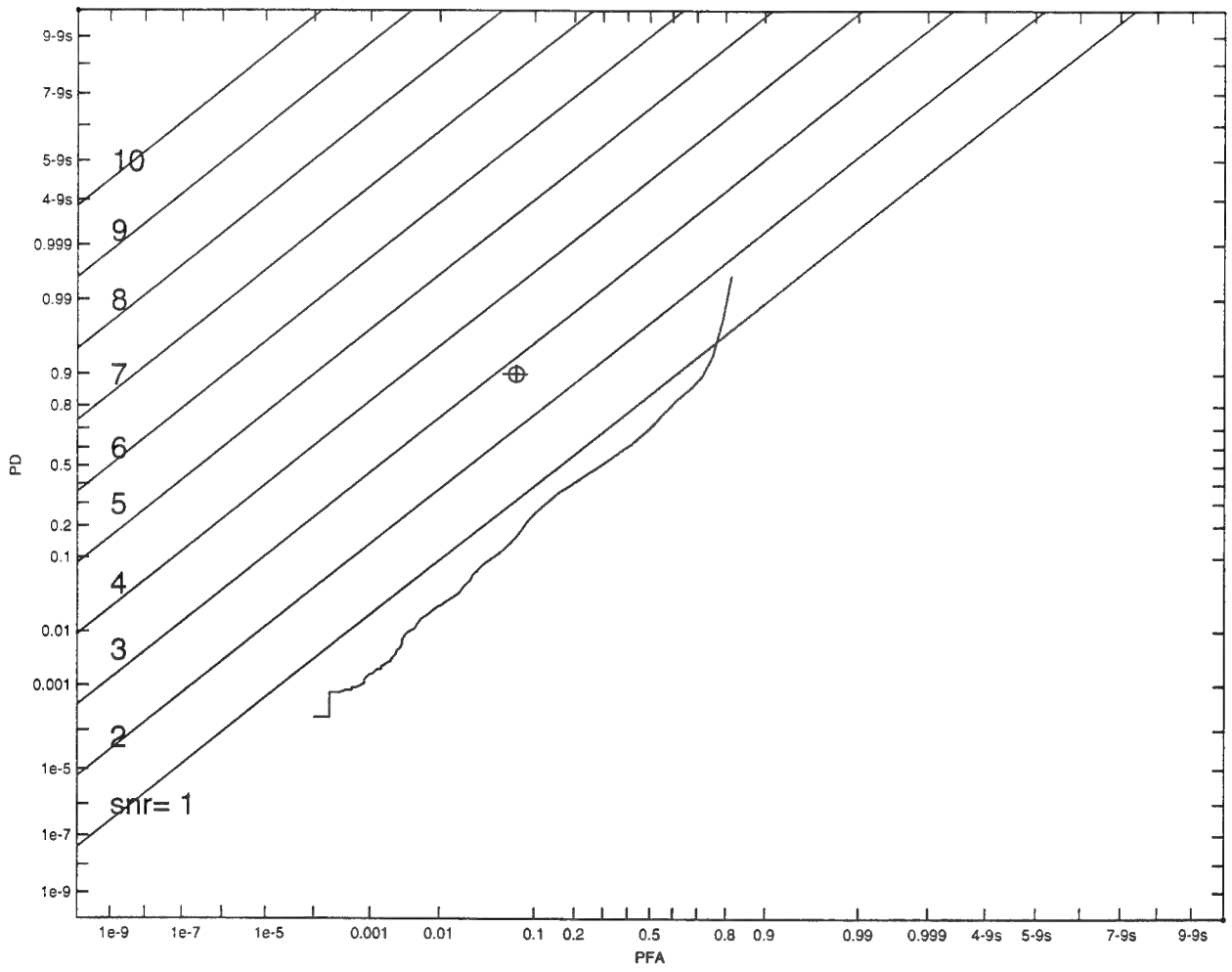


FIGURE 33. Experiment 2 Matched Filter Signal=4.5, Diameter=1.

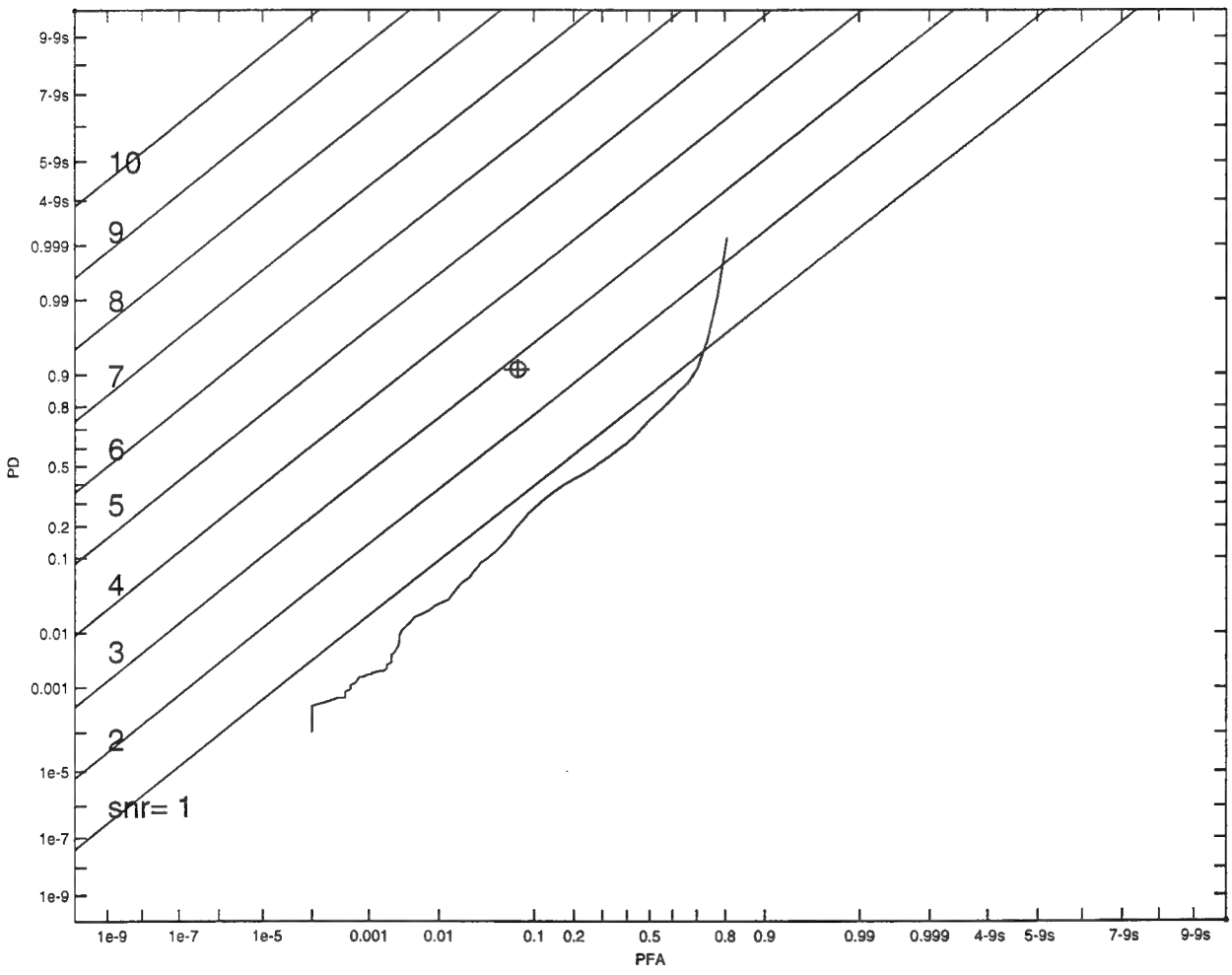


FIGURE 34. Experiment 2 Matched Filter Signal=4.5, Diameter=11

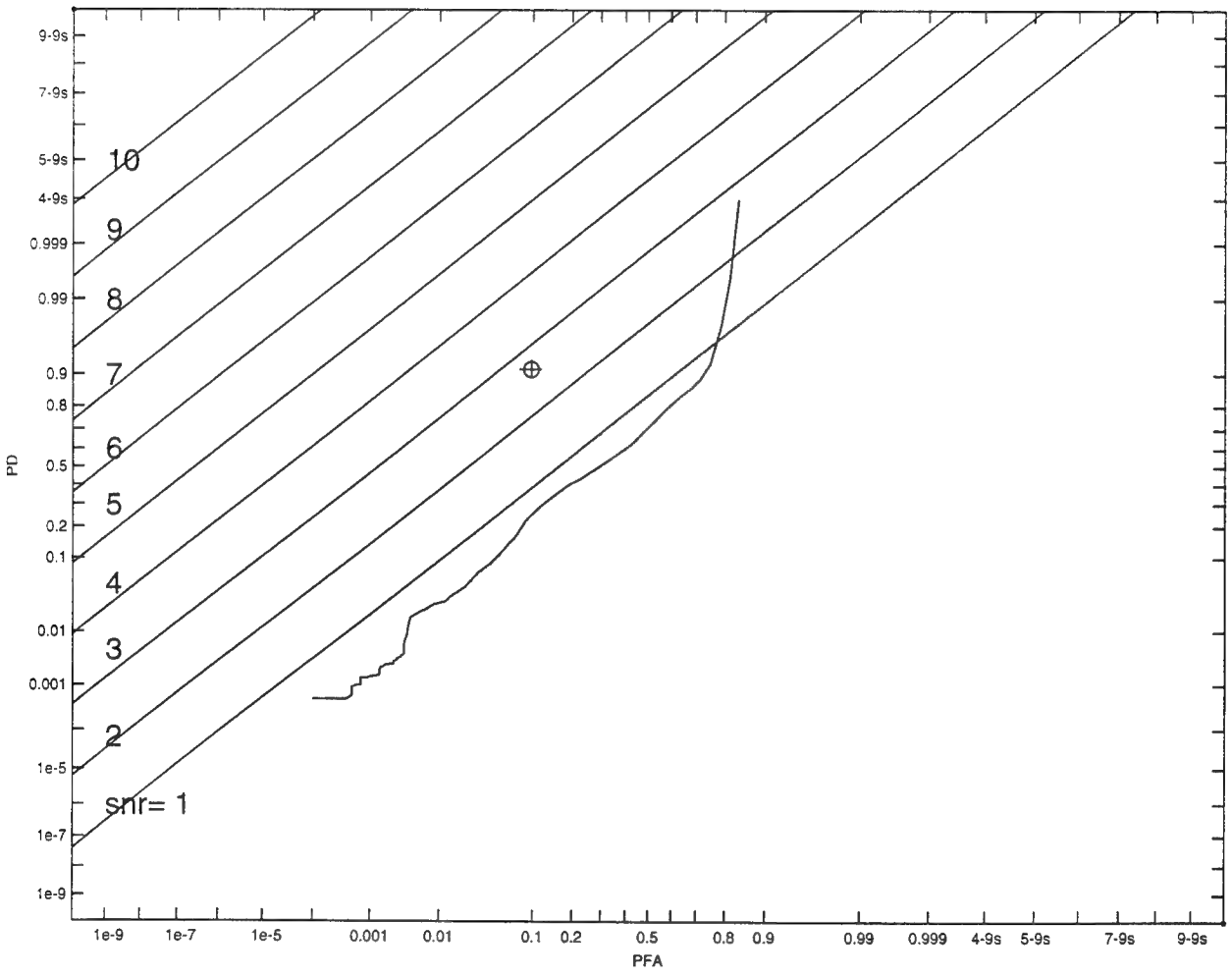


FIGURE 35. Experiment 2 Matched Filter Signal=4, Diameter=11.

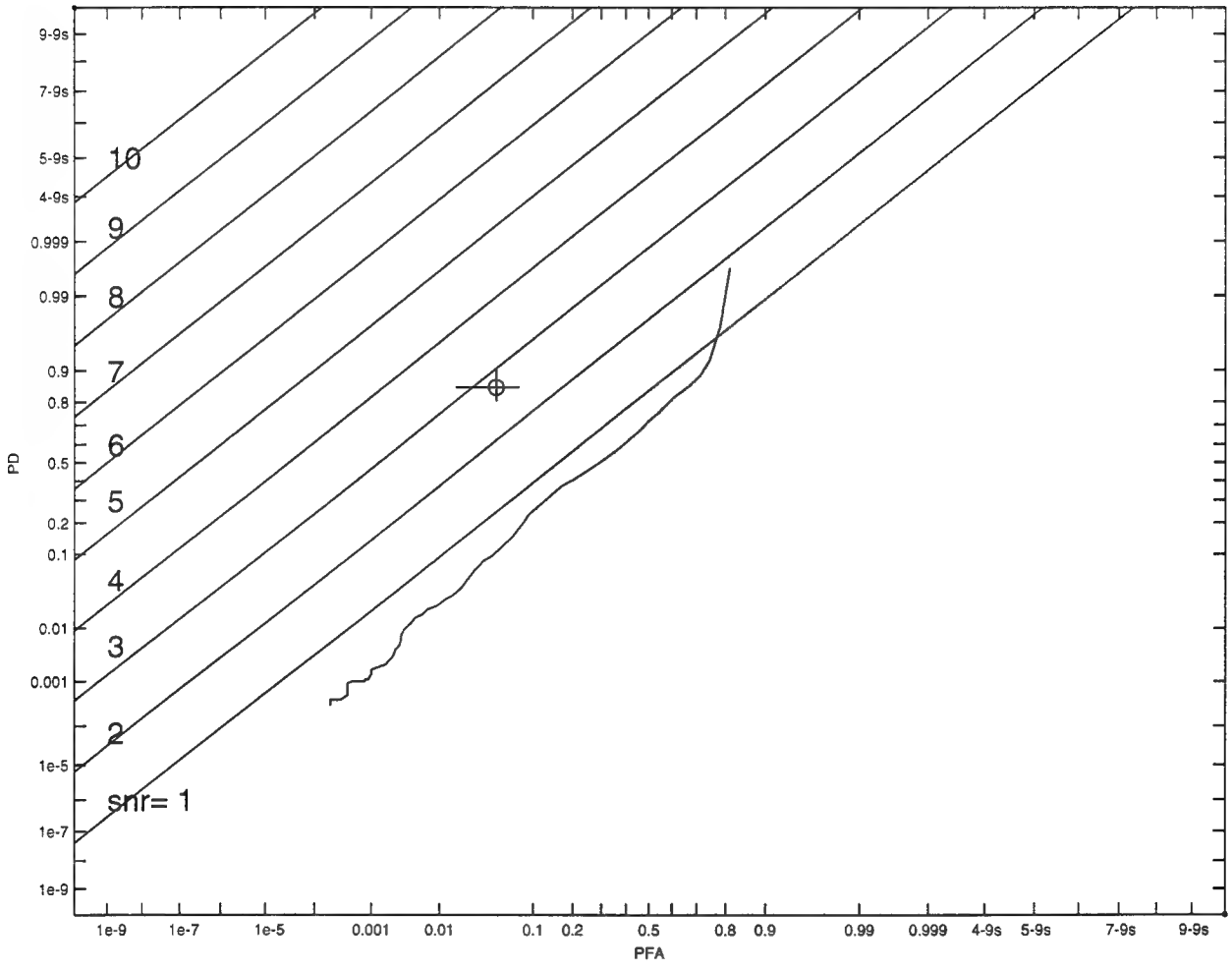


FIGURE 36. Experiment 2 Matched Filter Signal= 4.0, Diameter=3.

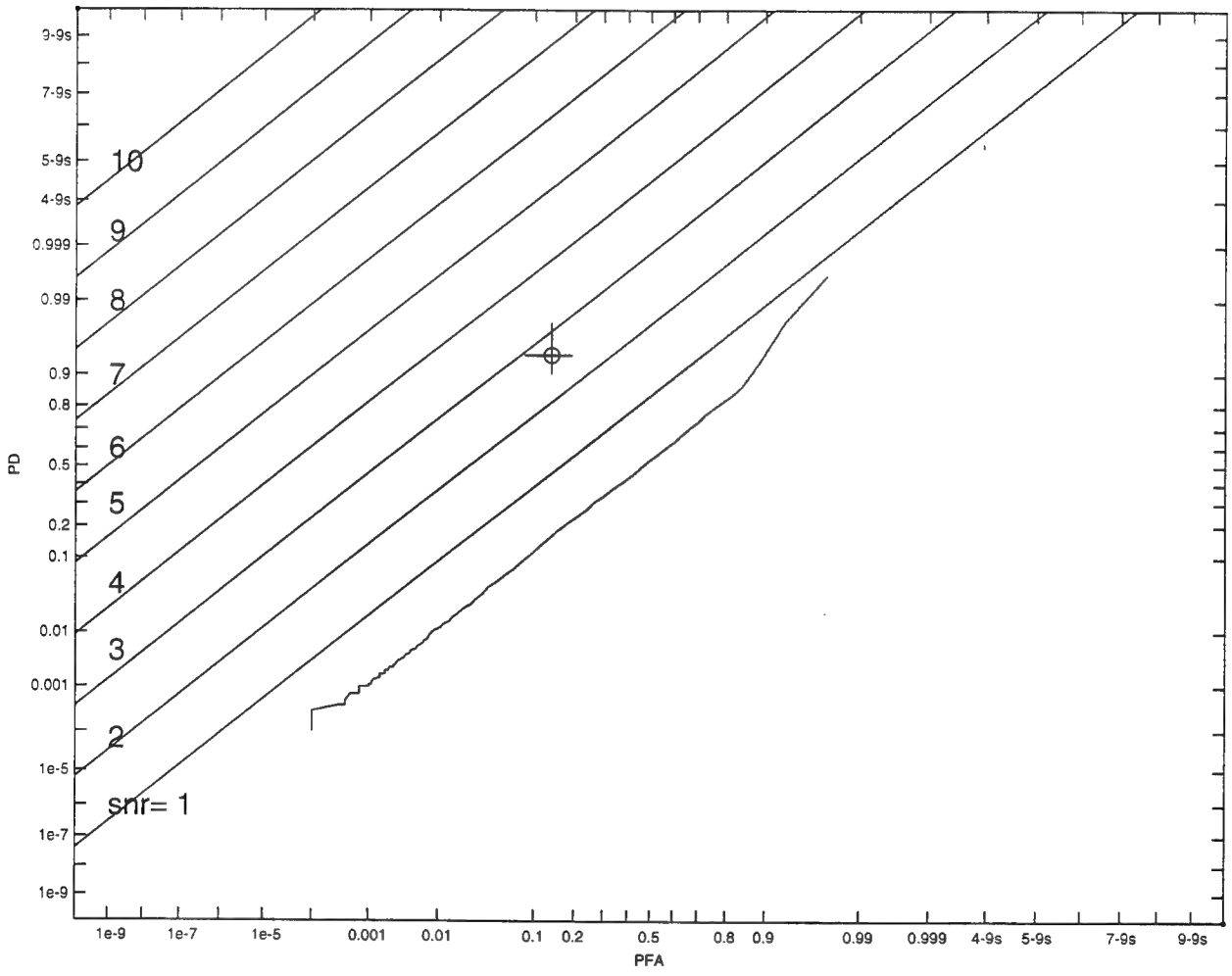


FIGURE 37. Experiment 2 Signal=4.0, Diameter=5.

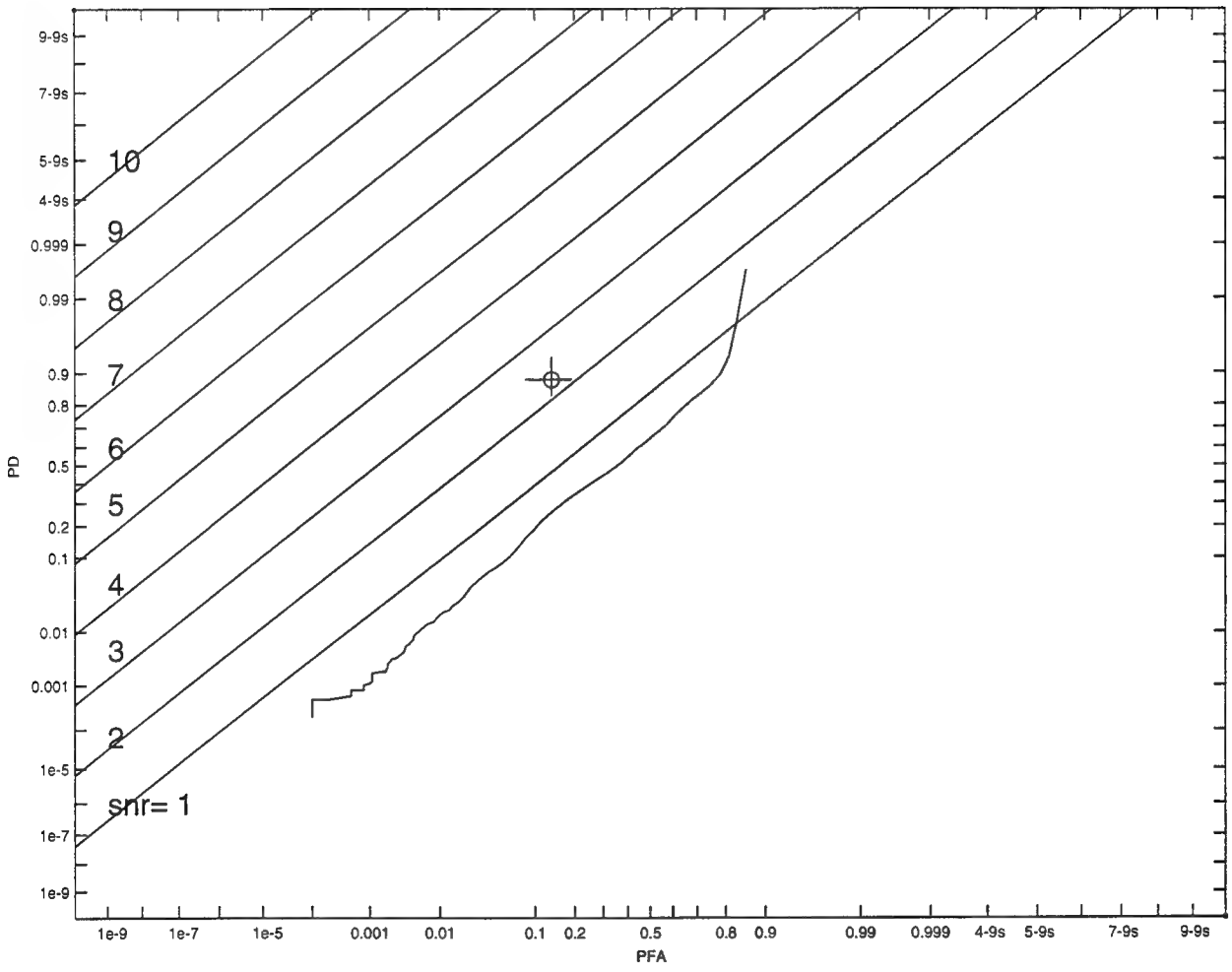


FIGURE 38. Experiment 2 Matched Filter Signal=4.5, Diameter=9.

3. EXPERIMENTAL DATA. EXPERIMENT 3

All experimental data for the optimum linear filter used in Experiment 3 are included in the main body of the thesis. The following pages show PRBPRB plots of an operator (P_D,P_{FA})-pair and a matched filter ROC.

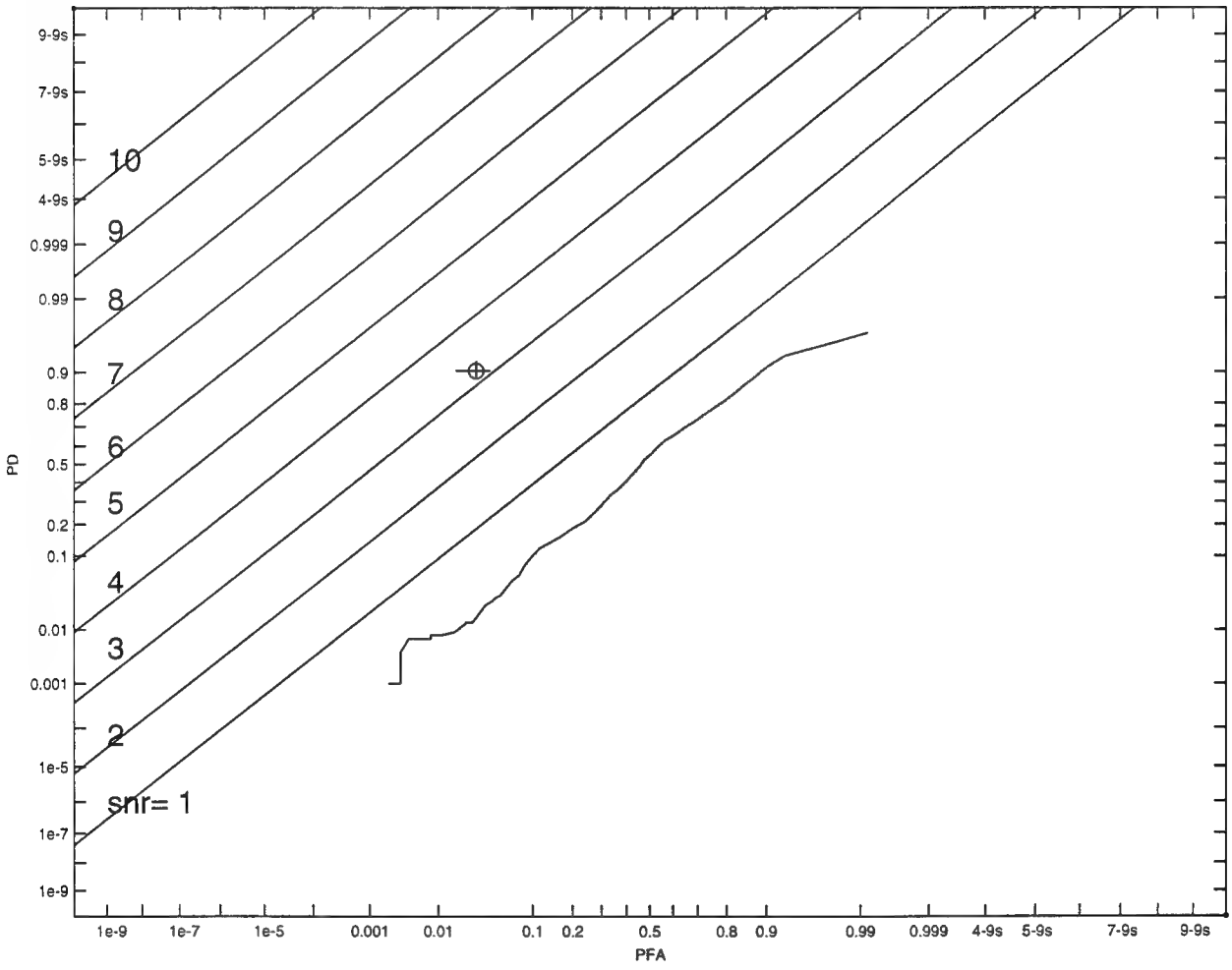


FIGURE 39. Experiment 3 Matched Filter Signal=0.0, Diametr=21

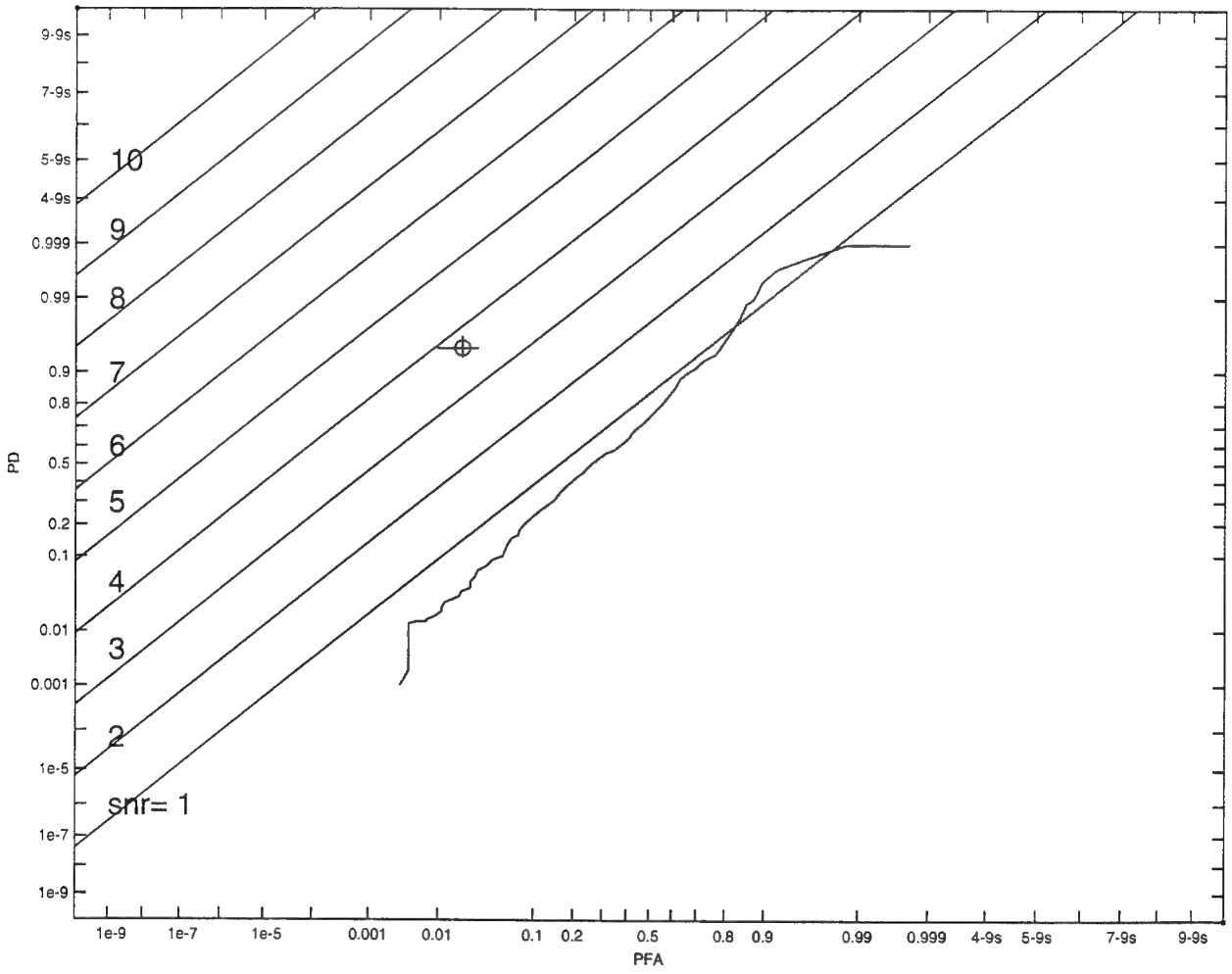


FIGURE 40. Experiment 3 Matched Filter Signal=1.5, Diameter=21.

APPENDIX C. COMPUTER PROGRAMS

This appendix contains the computer programs used in the experiments conducted for the thesis and a short discussion of the program flow for each experiment. Each eperiment was under the overall control of the computer. The computer acted as the control. center. In this command role the computer selected the background/target definitions and parameters. The computer made the decision of whether a target was to be present in a given display. Only the computer knew this information. Neither the operator nor the machine/filter were told whether or not a target was present. Each of the experiments collects both operator and machine performance data in the form of (P_D,P_{FA})-pairs. The programs used for this collection process specific to each experiment are discussed in subsections of this appendix. The last subsection contains the programs that are common to all experiments. The appendix contains four subsections. These subsections are:

1 . Appendix C-RWG Contains the programs that support Experiment 1.
2 . Appendix C-SPOT Contains the programs that support Experiment 2.
3 . Appendix C-SPOTC Contains the programs that supportExperiment 3.
4 . Appendix C-Common Contains the programs that are common to all the experiments.

Each experiment generated images for a human to observe on the computer screen. The human made a target present or not present decision through the use of the keyboard. The computer told the operator whether each decision he made was right or wrong and recorded these decisions as detections or false alarms. At the end of a sequence of trials the computer calculated probability of detection and probability of false alarm for the operator. For the operator the experimental data thus recorded was a single (P_D,P_{FA})-pair. The computer used the same target definitions and parameters to generate probabilities of detection and false alarm for a machine/filter equipped with either an optimum linear filter or for a matched filter. For the machine/filter the computer used 10,000 trials to generate (P_D,P_{FA})-pairs at 100 different thresholds using 100 different threshold settings. The computer collected 100 (P_D,P_{FA})-pairs for the machine/filter in each experiment.

The Programming language used in all of the programs was MATLAB.

APPENDIX C RWG

This appendix contains a printout of the programs written which support experiment 1. This experiment uses random white Gaussian (RWG) background images. The control program is OPMACH. This program calls two other programs called OPWHITE and MACHPERF. The OPWHITE program collects the operator (P_D, P_{FA})-pair and the MACHPERF program collects the machine (P_D, P_{FA})-pairs.

The OPWHITE program interfaces with the operator through the keyboard and the computer monitor. The computer monitor displays eighteen target/background images, and a decision box for the operator to input his choice of target present or not present. OPWHITE calls the programs MAKEPILL, SNRESTER and EXOPR3. The EXOPR3 program serves as the interface between the operator and the computer monitor for the experimental data collection phase of the experiment. This program collects the effective (P_D, P_{FA})-pair. EXOPR3 calls the programs CHECK, WRITEFILE, CAMO, SHOW, SCALE(1,2,3), CIRCDAT. The program CIRCDAT generates the target/background image. The MACHPERF program contains the algorithm to implement the machine/filter. The MACHPERF program in effect offers the same target/background definitions and parameters to the machine/filter as those shown to the operator. The MACHPERF program calls the programs MAKEPILL and NOISEGEN. The program NOISEGEN calculates the background pixel values needed such that a target with a specified signature will yield the nominal signal-to-noise-ratio.

EXPERIMENT 1

OPMACH

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called opmach designed to run opwhite and machperf.
% The program obtains operator and machine probability of detection,
% probability of false alarm data (PD,PFA)-pairs and graphs them for
% comparison.
%
%
%  INPUTS
% sig   (1,1)     = Input pixel value of target
% snrnom(1,1)     = Nominal signal to noise for experiment
% D     (1,1)     = Diameter of circular disc to be used in defining
%                   the filter; D must be odd.
% n     (1,1)     = number of displays
% Nsub  (1,num)   = number of pixels on the x axis.
%
%
%  OUTPUTS
% PDO    (1,1)    = Probability the operator had a true detection
% PFAO   (1,1)    = Probability the operator thought there was a
%                   target but there was none
% PDMOS (100,1)   = Probability of true detection-report by optimum
%                   filter at operator signal
% PFAMOS(100,1)   = Probability of false detection-report by optimum
%                   filter at operator  signal
%
%
%  [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig,snrnom)


function [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig,snrnom);
G=randn(256,256);
load start
[PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opwhite(n,sig,snrnom,D,Nsub);
[PDMOS,PFAMOS]=machperf(sig,snrnom,D,G,PDO,PFAO,Nsub);
```

```
load start;
H=prbprb(PFA,PD);
prprlaer(PDO,PFAO,H,'*');
prprlaer(PDMOS,PFAMOS,H)
load fnum;
fig=sprintf('exp#%1.0f',fnum)
```

OPWHITE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called opwhite designed to check an operator's nominal
% signal-to-noise ratio (snrnom) during a target recognition sequence.
% The function calculates an operators effective signal-to-noise-ratio
% (snreff) from the error function and the experimental (PD,PFA)-pair.
% The program then compares snreff with snrnom to ensure snreff
% is within 2 standard deviations of snrnom. The program allows the
% operator as many training trials as he requests and allows the
% operator to adjust the signal strength.
%
%
%  INPUTS
%   n    (1,1) =Total number of operator trials used in the experiment
% sig    (1,1) =Target signal level.
% D      (1,1) =Target radius
% Nsub   (1,1) =Size of background matrix to display
% snrnom(1,1) =Nominal signal to noise ratio
%
%
%  OUTPUTS
% snrexp(1,1) =Signal to noise ratio that the operator performs with
%              during the experiment.
% sigsnr(1,1) =Standard deviation of the snrnom
% snrbar(1,1) =average snr expected
% PDO    (1,1) =operator probability of detection
% PFAO   (1,1) =operator probability of false alarm
% c1     (2,1) = number of times target was present and
%                detected
% c2     (2,1) = number of times target was present and not
%                detected
% c3     (2,1) = number of times no target and no detection
% c4     (2,1) = number of times no target and false detection
%
%
% [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opwhite(n,sig,snrnom, ...
                                                    D,Nsub);


function [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]= ...
                                  opwhite(n,sig,snrnom,D,Nsub);
noise=noigen(snrnom,sig,D,Nsub);
[sigsnr,snrbar]=snrester(n,snrnom);
pill=makepill(Nsub,D);
[c1,c2,c3,c4,flag,PDO,PFAO,snrexp]=exopr3(n,Nsub,D,sig, ...
                                     noise,snrbar,sigsnr,pill);


```

EXOPR3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  function called exopr3 used to collect  operator (PD,PFA)-pairs.
% The program displays three series of target images. The first series
% displays a circular disc target in a background pattern. The second
% series displays a circular disc target in a background pattern where
% the background and target are between the minimum pixel value and the
```

```
% mean value of the pixels. The third series displays a circular disk
% target in a background pattern where the disc and the background
% pattern are between the mean background pixel value and the maximum
% background pixel value.  The monitor display provides the operator
% with the equivalent of a color/contrast control knob.
%
%             INPUTS
%    n        (1,1)       = number of displays
%   Nsub      (1,1)       = size of the background scene for display.
%    D        (1,1)       = circle Diameter
%   sig       (1,1)       = Target signal
%  noise      (1,1)       = Target noise
%  snrnom     (1,1)       = nominal signal to noise voltage ratio
%  sigsnr     (1,1)       = standard deviation of the signal to noise
%                           ratio
%    ss     (Nsub,Nsub)   = Box containing signal disc at center
%
%
%             OUTPUTS
%    c1       (2,1)       = number of times target was present and
%                           detected
%    c2       (2,1)       = number of times target was present and not
%                           detected
%    c3       (2,1)       = number of times no target and no detection
%    c4       (2,1)       = number of times no target and false detection
%    flag     (1,1)       = emergency exit indicator,
%                           (1/0)=(e-exit/norm-end)
%    PDO      (1,1)       = Probability the target was present and the
%                           operator identified it
%    PFAO     (1,1)       = Probability the target was not present but
%                           the operator thought it was present
%  snrexp     (1,1)       = Signal to noise voltage ratio of the operator
%                           during the experiment
%
%
%   [c1,c2,c3,c4,flag,PDO,PFAO,snrexp]=exopr3(n,Nsub,D,sig, ...
                                          noise,snrnom,sigsnr,ss)

function [c1,c2,c3,c4,flag,PDO,PFAO,snrexp]=exopr3(n,Nsub,D,sig, ...
            noise,snrnom,sigsnr,ss);
R=(D-1)/2;
c1=0;c2=0;c3=0;c4=0;
cla,clg,axis('off');
flag=0;
count=1;
for k=1:n
   if flag == 1 end
        T=(rand(1,1)>0.5);
   cla,clg;
   kx=rand(1,1);
   while kx==1;
      kx=rand(1,1);
   end
   ky=rand(1,1);
   while ky==1;
      ky=rand(1,1);
   end
   [mm,SD,zp1]=circdat(Nsub,D,noise,sig);
   zs1=zp1+(ss*sig*T);
   subl1=scaling1(zs1,mm,SD,.01,R);
   showr(subl1,.025,.05,.14,.3,(D-1)/2);
   subl1=scaling1(zs1,mm,SD,1,R);
   showr(subl1,.19,.05,.14,.3,(D-1)/2);
   subl1=scaling1(zs1,mm,SD,1.5,R);
   showr(subl1,.355,.05,.14,.3,(D-1)/2);
   subl1=scaling1(zs1,mm,SD,2,R);
   showr(subl1,.52,.05,.14,.3,(D-1)/2);
   subl1=scaling1(zs1,mm,SD,2.5,R);
```

```
75      showr(subl1,.685,.05,.14,.3,(D-1)/2);
76      subl1=scaling1(zs1,mm,SD,3,R);
77      showr(subl1,.85,.05,.14,.3,(D-1)/2);
78
79      [mm1,SD1,zp2]=circdat(Nsub,D,noise,sig);
80      zs2=zp2+(ss*sig*T);
81      [subl2,scale]=scaling2(zs2,mm1,SD1,1.4,R);
82      showr(subl2,.025,.35,.14,.3,(D-1)/2);
83      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+1),R);
84      showr(subl2,.19,.35,.14,.3,(D-1)/2);
85      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+2),R);
86      showr(subl2,.355,.35,.14,.3,(D-1)/2);
87      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+3),R);
88      showr(subl2,.52,.35,.14,.3,(D-1)/2);
89      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+4),R);
90      showr(subl2,.685,.35,.14,.3,(D-1)/2);
91      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+5),R);
92      showr(subl2,.85,.35,.14,.3,(D-1)/2);
93
94      [mm2,SD2,zp3]=circdat(Nsub,D,noise,sig);
95      zs3=zp3+(ss*sig*T);
96      [subl3,scale]=scaling3(zs3,mm2,SD2,1,R);
97      showr(subl3,.025,.65,.14,.3,(D-1)/2);
98      [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+.5),R);
99      showr(subl3,.19,.65,.14,.3,(D-1)/2);
100     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+.75),R);
101     showr(subl3,.355,.65,.14,.3,(D-1)/2);
102     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
103     showr(subl3,.52,.65,.14,.3,(D-1)/2);
104     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1.5),R);
105     showr(subl3,.685,.65,.14,.3,(D-1)/2);
106     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+2),R);
107     showr(subl3,.85,.65,.14,.3,(D-1)/2);
108     disp('target number is');
109     disp (count);
110        tgt = input('is there a target? [(y/n); ~(y/n) ...
111                exits test loop] >>','s');
112          if (tgt=='y'| tgt=='n')
113        count=count+1;
114        if T;
115           if tgt=='y'
116              disp('CORRECT');
117              c1=c1+1;
118           else
119              disp('WRONG');
120              c2=c2+1;
121           end
122        else
123           if tgt=='y'
124              disp('WRONG');
125              c4=c4+1;
126           else
127              disp('CORRECT');
128              c3=c3+1;
129           end
130        end
131        if count==floor(n/5) | count==floor(2*n/5) | count== floor(3*n/5) ...
132              | count==floor(4*n/5) |count==n;
133           [PDO,PFAO,snrexp,flag]=check(c1,c2,c3,c4,snrnom,sigsnr);
134           save opwdata c1 c2 c3 c4 PDO PFAO snrexp sigsnr sig snrnom D Nsub;
135           writefil(count,c1,c2,c3,c4,snrexp,PDO,PFAO,sigsnr,sig,snrnom,D,Nsub);
136        else
137        PDO=c1/(c1+c2);
138        PFAO=c4/(c3+c4);
139           if PDO<1 & PDO>0 & PFAO<1 &PFAO>0
140              t2sig=erfinv(1-2*PFAO);
141              snrexp=sqrt(2)*(t2sig-erfinv(1-2*PDO));
142              sprintf('PDO:%5.4f PFAO:%5.4f snrexp:%5.4f',PDO,PFAO,snrexp)
```

```
            end
         end
         if flag==1 break; end
      else
         flag=1;
         break;
      end
   end
   PD0=(c1)/(c1+c2);
   PFA0=(c4)/(c3+c4);
   t2sig=erfinv(1-2*PFA0);
   snrexp=sqrt(2)*(t2sig-erfinv(1-2*PD0));
```

CIRCDAT

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called circdat used to generate a pixel image of a circle
% in a square background.  The background pixel values have a gaussian
% random distribution
%
%           INPUTS
%  Nsub    (1,1)      = number of pixels on the side of the
%                       square
%  D       (1,1)      = circle diameter
%  noise   (1,1)      = rms noise level
%  signal  (1,1)      = fixed level (above or below one-half)
%                       for circle signal
%
%            OUTPUTS

%   mm     (1,1)      = mean pixel value of target circle
%  SD      (1,1)      = pixel standard deviation in sub with target
%  pic   (Nsub,Nsub) = Random background matrix
%
%
%     [mm,SD,pic]=circdat(Nsub,D,noise,signal)


function [mm,SD,pic]=circdat(Nsub,D,noise,signal);
R=(D-1)/2;
pic=noise*randn(Nsub,Nsub)+0.5;
x=((1:Nsub)-Nsub/2).^2; x=x'*ones(size(x)); y=x';
xy=x+x';
ii=find(xy>R^2);
jj=find(xy<=R^2);
jjj=find(xy==R^2);
Sig=signal*ones(Nsub,Nsub);
Sig(ii)=zeros(size(Sig(ii)));
pic=pic+Sig;
Tu=0.5+signal+2.0*noise;
Tl=0.5-2.0*noise;
Cout=sum(sum(ones(size(ii))));
C=Nsub^2-Cout;
kk=find(pic>Tu);
picu=zeros(size(pic));
picu(kk)=pic(kk)-Tu;
pic(kk)=Tu*ones(size(pic(kk)));
ll=find(pic<Tl);
picl=zeros(size(pic));
picl(ll)=pic(ll)-Tl;
pic(ll)=Tl*ones(size(pic(ll)));
picul=picu+picl;
adjout=sum(sum(picul(ii)))/Cout;
pic(ii)=pic(ii)+adjout;
adjin=sum(sum(picul(jj)))/C;
pic(jj)=pic(jj)+adjin;
pic(1,1)=Tu+0.25*noise;
```

```
53  pic(1,2)=T1-0.25*noise;
54  mm=mean(pic(jj));
55  SD=std(pic(jj(:)));
```

MACHPERF

```
 1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2  % Function called machperf designed to analyze performance in terms
 3  % of PD/PFA of a machine in looking for a uniform circular disc target.
 4  % The target is present in a background scene that is randomly chosen.
 5  % The machine uses a set of filter weights (from a matched filter)
 6  % and a threshold to detect targets.
 7  %
 8  %
 9  %  INPUTS
10  % sig   (1,1)       = Input pixel value of target
11  % snrnom(1,1)       = nominal signal to noise ratio
12  % D    (1,1)        = Diameter of circular disc to be used in defining
13  %                     the filter; D must be odd.
14  % G    (256,256)    = Random background
15  % PDO    (1,1)      = Operator probability of detection
16  % PFAO   (1,1)      = Operator probability of false alarm
17  % Nsub   (1,N)      = Size of background scene to display
18  %
19  %  OUTPUTS
20  % PDMOS  (100,1)    = Probability of true detection-report by optimum
21  %                     filter at operator signal
22  % PFAMOS(100,1)     = Probability of false detection-report by optimum
23  %                     filter at operator signal
24  %
25  %
26  %
27  %   [PDMOS,PFAMOS]=machperf(sig,snrnom,D,G,PDO,PFAO,Nsub);
28
29  function  [PDMOS,PFAMOS]=machperf(sig,snrnom,D,G,PD,PFA,Nsub);
30  sigm=sig;
31  noise=noigen(snrnom,sig,D,Nsub);
32  pill=makepill(Nsub,D);
33  R=(D-1)/2;
34  RN=(Nsub-1)/2;
35  x=(-RN:RN).^2;
36  x=ones(size(x'))*x;
37  R2=x+x';
38  ii=find(R2<=R^2);
39  L=length(ii);
40  pd=256-Nsub;
41  qd=256-Nsub;
42  Sum1=[];
43  N=10000;
44  X=[];
45  Y=[];
46  A=sum(pill(ii));
47  for b=1:N
48     x= (RN+1)+pd*rand(1,1);
49     y= (RN+1)+qd*rand(1,1);
50     Z= G(x-RN:x+RN,y-RN:y+RN);
51     Sum1=[Sum1; sum(sum(Z.*pill))];
52  end
53  Sums=[Sum1*noise Sum1*noise+sigm*L];
54  T=linspace(min(min(Sums)),max(max(Sums)),102);
55  T(102)=[];
56  T(1)=[];
57  PDMOS=[];
58  PFAMOS=[];
59  for b=1:100
60     e=(T(b) < Sums);
61     e=sum(e);
```

```
62      PDMOS=[PDMOS e(2)/N];
63      PFAMOS=[PFAMOS e(1)/N];
64  end
```

APPENDIX C SPOT

This appendix contains a printout of the programs written which support experiment 2. This experiment uses SPOT background images. The control program is OPMACH. This program calls programs called OPNWHITE and MACHPERF. The OPNWHITE program collects the operator (P_D,P_{FA})-pair and the MACHPERF program collects the machine (P_D,P_{FA})-pairs.

The OPNWHITE program provides the operator with a training phase then when the operator feels comfortable with target recognition the program begins the experimental phase. The OPNWHITE program interfaces with the operator through the keyboard and the computer monitor. The computer monitor displays eighteen target/background images, and a decision box for the operator to input his choice of target present or not present. OPNWHITE calls the programs MAKEPILL, PDSNREST and EXOPR3. PDSNREST estimates the effective (P_D,P_{FA})-pair of an xperiment. PDSNREST allows the operator to have as many training trials as he requests. The program also allows the operator to adjust the target signature strength so that the effective (P_D,P_{FA})-pair is close to the nominal (P_D,P_{FA})-pair. The PDSNREST calls the program EXOPR3 which serves as the interface between the monitor and the operator and collects/calculates the operator effective (P_D,P_{FA})-pair. The EXOPR4 program serves as the interface between the operator and the computer monitor for the experimental data collection phase of the experiment. This program collects the effective (P_D,P_{FA})-pair collected as data. EXOPR4 calls the programs CHECK, WRITEFILE, CAMO, SHOW, SCALE(1,2,3), MTAR. MTAR adds a target of uniform signature to the background pixels. The MACHPERF program contains the algorithm to implement the machine/filter. The MACHPERF program in effect offers the same target/background definitions and parameters to the machine/filter as those shown to the operator. The MACHPERF program calls the programs PSDGEN, MAKEPILL, CAM. MAKEPILL and CAM serve the same purpose as in the operator collection phase. The program PSDGEN calculates the background matrix spatial frequency power spectral density (PSD) which is then used to implement the optimum linear filter.

EXPERIMENT 2

OPMACH

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called opmach designed to run opwhite and machperf.
% The program obtains operator and machine probability of detection,
% probability of false alarm and graphs them for comparison.
%
%
%  INPUTS
% sig   (1,1)    = Input pixel value of target
% D     (1,1)    = Diameter of circular disc to be used in defining the
%                  filter; D must be odd.
% n     (1,1)    = number of displays
% Nsub  (1,num)  = number of pixels on the x axis.
%
%
%  OUTPUTS
% PDO     (1,1)  = Probability the operator had a true detection
% PFAO    (1,1)  = Probability the operator thought there was a
%                  target but there was none
% PDMOS   (100,1)= Probability of true detection-report by optimum
%                  filter at operator signal
% PFAMOS(100,1) = Probability of false detection-report by optimum
%                  filter at operator  signal
%
%
%  [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig)

```

```
27
28 function [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig);
29 load start
30 [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opnwhite(n,sig,z,D,Nsub);
31 [PDMOS,PFAMOS]=machperf(sig,D,z,PDO,PFAO,Nsub,ssow);
32 H=prbprb(PFA,PD);
33 prbprb(PFAO,PDO,H,'ro');
34 prbprb(PFAMOS,PDMOS,H,'g-');
35 load fnum;
36 fig=sprintf('exp#%1.0f',fnum)
```

OPNWHITE

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 % Function called opnwhite designed to check an operator's  effective
 3 % signal-to-noise ratio (snreff) during a target recognition sequence.
 4 % The program calculates an operator's (snr) from the error function
 5 % and  the operator (PD,PFA)-pair. The program then compares this value
 6 % with the nominal snr (snrnom) to see if the operator snreff is within
 7 % two standard deviations of snrnom.  The program allows the operator as
 8 % many trials for trainup as he requests.  The program gives the
 9 % operator the capability to adjust the signal of the experiment.
10 %
11 %
12 %  INPUTS
13 % n    (1,1)    =Total number of operator trials used in the experiment
14 % sig  (1,1)    =Target signal level
15 % D    (1,1)    =Target radius
16 % Nsub(1,1)    =Size of background matrix to display
17 % z(256,256)   =background to display
18 %
19 %
20 %  OUTPUTS
21 % snrexp(1,1) =Signal to noise ratio that the operator performs with
22 %               during the experiment.
23 % sigsnr(1,1) =Standard deviation of the snrnom
24 % snrbar(1,1) =average snr expected
25 % PDO    (1,1) =operator probability of detection
26 % PFAO   (1,1) =operator probability of false alarm
27 %     c1 (2,1) =number of times target was present and
28 %               detected
29 %     c2 (2,1) =number of times target was present and not
30 %               detected
31 %     c3 (2,1) =number of times no target and no detection
32 %     c4 (2,1) =number of times no target and false detection
33 %
34 %
35 % [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opnwhite(n,sig,z,D,Nsub);
36
37
38
39 function [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]= ...
40                              opnwhite(n,sig,z,D,Nsub);
41 pill=makepill(Nsub,D);
42 [sigsnr,snrbar,sig]=pdsnrest(n,Nsub,D,sig,z,pill);
43 [c1,c2,c3,c4,flag,PDO,PFAO,snrexp]= ...
44                     exopr4(n,Nsub,D,sig,z,snrbar,sigsnr,pill);
```

MACHPERF

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 % Function called machperf designed to analyze performance in terms
 3 % of PD/PFA of a machine in looking for a uniform circular disc target
 4 % with imperfect, camouflage. The target is present in a background
 5 % scene that is randomly chosen.The machine uses a set of filter
 6 % weights to recognize targets.
 7 %
```

```
%  INPUTS
% sigm   (1,1)      = Input pixel value of target
% D      (1,1)      = Diameter of circular disc to be used in defining
%                     the filter; D must be odd.
% z   (256,256)     = Background scene
% PDO    (1,1)      = Operator probability of detection
% PFAO   (1,1)      = Operator probability of false alarm
% Nsub   (1,N)      = Size of background scene to display
% filt   (Nsub,Nsub)= filter weight array
%
%  OUTPUTS
% PDMOS   (100,1)   = Probability of true detection-report by optimum
%                     filter at operator signal
% PFAMOS(100,1)     = Probability of false detection-report by optimum
%                     filter at operator signal
%
%
%   [PDMOS,PFAMOS]=machperf(sigm,D,z,PDO,PFAO,Nsub,filt);

function  [PDMOS,PFAMOS]=machperf(sigm,D,z,PDO,PFAO,Nsub,filt);
R=(D-1)/2;
RN=(Nsub-1)/2;
x=(-RN:RN).^2;
x=ones(size(x'))*x;
R2=x+x';
ii=find(R2<=R^2);
L=length(ii);
if exist( 'ss');
   pill=makepill(Nsub,D);
   filt=pill;
else exist('ssow');
   pill=makepill(256,D);
   s=fft2(pill);
   PSD=psdgen(50,z);
   ssow=real(ifft2(s./PSD));
   P=(129-(Nsub-1)/2):(129+(Nsub-1)/2);
   ssow=ssow(P,P);
   filt=ssow;
end
pd=256-Nsub;
qd=256-Nsub;
Sum1=[];
N=10000;
X=[];
Y=[];
A=sum(filt(ii));
for b=1:N
   x= (RN+1)+pd*rand(1,1);
   y= (RN+1)+qd*rand(1,1);
   Z= z(x-RN:x+RN,y-RN:y+RN);
   Sum1=[Sum1; sum(sum(Z.*filt))];
end
Sums=[Sum1 Sum1+sigm*A];
T=linspace(min(min(Sums)),max(max(Sums)),102);
T(102)=[];
T(1)=[];
PDMOS=[];
PFAMOS=[];
for b=1:100
   e=(T(b) < Sums);
   e=sum(e);
   PDMOS=[PDMOS e(2)/N];
   PFAMOS=[PFAMOS e(1)/N];
end
```

PSDGEN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called psdgen used to develop a two dimensional power
% spectral density (psd) from a two dimensional background scene. Based
% on the assumption of isotropy a one dimensional result, psd versus
% spatial frequency magnitude, is developed. A single scene is used and
% ensemble averaging is "accomplished" by averaging over a ring in
% spatial frequency space. The program uses spline interpolation on the
% calculated set of psd values to generate a two dimensional array of
% psd values corresponding to the frequencies of the two dimensionnal
% FFT of the input background scene.
%
%
%  INPUTS
% N   (1,1)     = Number of spatial frequencies
% z   (256,256) = Background scene
%
%
%  OUTPUTS
% PSD (256,256) = 2-D version of psd
%
%    PSD=psdgen(N,z)

function PSD=psdgen(N,z);
zz=fft2(z);
k=0:127; k=[k -fliplr(k+1)];
k=ones(256,1)*k;
k=k.^2; k=k+k'; k=sqrt(k);
f=128*sqrt(2)*(1:N)/N;
df=(f(2)-f(1))/2;
psd=[];
for n=1:N
   ii= find(k>f(n)-df & k<=f(n)+df);
   psd = [psd sum((abs(zz(ii))).^2)/length(ii)];
end
psd(35)=(psd(34)+psd(36))/2;
k=k(:);
PSD=spline(f,psd,k);
PSD=reshape(PSD,256,256);
```

EXOPR3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  function called exopr3 used to collect operator (PD,PFA)-pairs.
% The program displays three series of target images. The first series
% displays a circular disc target in a background pattern. The second
% series displays a circular disc target in a background pattern where
% the background and target are between the minimum pixel value and the
% mean value of the pixels. The third series displays a circular disk
% target in a background pattern where the disc and the background
% pattern are between the mean background pixel value and the maximum
% background pixel value. The monitor display is designed to provide
% the operator with the equivalent of a color/contrast control knob.
% The function collects operator performance numbers from which pd and
% pfa are calculated. The data collection is based on some specified
% number of separate displays. This program is designed for the
% pre-experiment stage when the operator is being trained and PD,PFA
% are being verified.
%
%             INPUTS
%    n        (1,1)       = number of displays
%   Nsub      (1,1)       = size of the background scene for display.
%    D        (1,1)       = circle Diameter
%   sig       (1,1)       = Target signal
%    z        (256,256)   = Random image field
%    ss       (Nsub,Nsub)= Box containing signal disc at center
%
```

```
26 %
27 %              OUTPUTS
28 %    c1        (2,1)       = number of times target was present and
29 %                            detected
30 %    c2        (2,1)       = number of times target was present and not
31 %                            detected
32 %    c3        (2,1)       = number of times no target and no detection
33 %    c4        (2,1)       = number of times no target and false detection
34 %    flag      (1,1)       = emergency exit indicator,
35 %                            (1/0)=(e-exit/norm-end)
36 %    PD         (1,1)      = Probability the target was present and the
37 %                            operator identified it
38 %    PFA       (1,1)       = Probability the target was not present but
39 %                            the operator thought it was present
40 %
41 %
42
%  [c1,c2,c3,c4,flag,PD,PFA]=exopr3(n,Nsub,D,sig,z,ss)
43
44
45
46 function [c1,c2,c3,c4,flag,PD,PFA]=exopr3(n,Nsub,D,sig,z,ss);
47 if n>25 n=25; end;
48 R=(D-1)/2;
49 c1=0;c2=0;c3=0;c4=0;
50 cla,clg,axis('off');
51 flag=0;
52 count=1;
53 for k=1:n
54    if flag == 1 end
55         T=(rand(1,1)>0.5);
56    cla,clg;
57    kx=rand(1,1);
58    while kx==1;
59       kx=rand(1,1);
60    end
61    ky=rand(1,1);
62       while ky==1;
63    ky=rand(1,1);
64    end
65    [mm,SD,zp1]=mtar(Nsub,D,z,kx,ky);
66    zs1=zp1+(ss*sig*T);
67    subl1=scaling1(zs1,mm,SD,.01,R);
68    showr(subl1,.025,.05,.14,.3,(D-1)/2);
69    subl1=scaling1(zs1,mm,SD,1,R);
70    showr(subl1,.19,.05,.14,.3,(D-1)/2);
71    subl1=scaling1(zs1,mm,SD,1.5,R);
72    showr(subl1,.355,.05,.14,.3,(D-1)/2);
73    subl1=scaling1(zs1,mm,SD,2,R);
74    showr(subl1,.52,.05,.14,.3,(D-1)/2);
75    subl1=scaling1(zs1,mm,SD,2.5,R);
76    showr(subl1,.685,.05,.14,.3,(D-1)/2);
77    subl1=scaling1(zs1,mm,SD,3,R);
78    showr(subl1,.85,.05,.14,.3,(D-1)/2);
79
80    [mm1,SD1,zp2]=mtar(Nsub,D,z,kx,ky);
81    zs2=zp2+(ss*sig*T);
82    [subl2,scale]=scaling2(zs2,mm1,SD1,1.4,R);
83    showr(subl2,.025,.35,.14,.3,(D-1)/2);
84    [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+1),R);
85    showr(subl2,.19,.35,.14,.3,(D-1)/2);
86    [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+2),R);
87    showr(subl2,.355,.35,.14,.3,(D-1)/2);
88    [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+3),R);
89    showr(subl2,.52,.35,.14,.3,(D-1)/2);
90    [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+4),R);
91    showr(subl2,.685,.35,.14,.3,(D-1)/2);
92    [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+5),R);
```

```
93      showr(subl2,.85,.35,.14,.3,(D-1)/2);
94
95      [mm2,SD2,zp3]=mtar(Nsub,D,z,kx,ky);
96      zs3=zp3+(ss*sig*T);
97      [subl3,scale]=scaling3(zs3,mm2,SD2,.5,R);
98      showr(subl3,.025,.65,.14,.3,(D-1)/2);
99      [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
100     showr(subl3,.19,.65,.14,.3,(D-1)/2);
101     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
102     showr(subl3,.355,.65,.14,.3,(D-1)/2);
103     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
104     showr(subl3,.52,.65,.14,.3,(D-1)/2);
105     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
106     showr(subl3,.685,.65,.14,.3,(D-1)/2);
107     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
108     showr(subl3,.85,.65,.14,.3,(D-1)/2);
109
110
111     disp('target number is');
112     disp (count);
113     tgt = input('is there a target? [(y/n); ~(y/n) exits test loop] >>','s');
114          if (tgt=='y'| tgt=='n')
115     count=count+1;
116        if T;
117           if tgt=='y'
118              disp('CORRECT');
119              c1=c1+1;
120           else
121              disp('WRONG');
122              c2=c2+1;
123           end
124        else
125           if tgt=='y'
126              disp('WRONG');
127              c4=c4+1;
128           else
129              disp('CORRECT');
130              c3=c3+1;
131           end
132        end
133     else
134        flag=1;
135        break;
136     end
137  end
138  PD=(c1)/(c1+c2);
139  PFA=(c4)/(c3+c4);
```

EXOPR4

```
1   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2   % function called exopr4 used to collect pd/pfa related data. This
3   % function will display three series of target images. The first series
4   % displays a  circular disc target in a background pattern. The second
5   % series displays a circular disc target in a background pattern where
6   % the background and target are between the minimum pixel value and the
7   % mean value of the pixels. The third series displays a circular disk
8   % target in a background pattern where the disc and the background
9   % pattern are between the mean background pixel value and the maximum
10  % background pixel value.  The monitor display is designed to provide
11  % the operator with the equivalent of a color/contrast control knob.
12  % The function collects operator performance numbers from which pd and
13  % pfa are calculated. The data collection is based on some specified
14  % number of separate displays. This program is designed for the
15  % experiment stage when the operator (PD,PFA)-pair is collected.
16  %
17  %             INPUTS
```

```
18 %    n         (1,1)       = number of displays
19 %   Nsub       (1,1)       = size of the background scene for display.
20 %    D         (1,1)       = circle Diameter
21 %   sig        (1,1)       = Target signal
22 %    z        (256,256)    = Background scene to display
23 %  snrnom      (1,1)       = nominal signal to noise voltage ratio
24 %  sigsnr      (1,1)       = standard deviation of the signal to noise
25 %                             ratio
26 %    ss      (Nsub,Nsub)   = Type of filter
27 %
28 %
29 %              OUTPUTS
30 %    c1        (2,1)       = number of times target was present and
31 %                             detected
32 %    c2        (2,1)       = number of times target was present and not
33 %                             detected
34 %    c3        (2,1)       = number of times no target and no detection
35 %    c4        (2,1)       = number of times no target and false detection
36 %    flag      (1,1)       = emergency exit indicator,
37 %                             (1/0)=(e-exit/norm-end)
38 %    PD0       (1,1)       = Probability the target was present and the
39 %                             operator identified it
40 %    PFA0      (1,1)       = Probability the target was not present but
41 %                             the operator thought it was present
42 %   snrexp     (1,1)       = Signal to noise voltage ratio of the
43 %                             operator during the experiment
44 %
45 %
46 %   [c1,c2,c3,c4,flag,PD0,PFA0,snrexp]=exopr4(n,Nsub,D,sig, ...
47                 z,snrnom,sigsnr,ss)
48
49 function [c1,c2,c3,c4,flag,PD0,PFA0,snrexp]=exopr4(n,Nsub,D,sig, ...
50              z,snrnom,sigsnr,ss);
51 R=(D-1)/2;
52 c1=0;c2=0;c3=0;c4=0;
53 cla,clg,axis('off');
54 flag=0;
55 count=1;
56 for k=1:n
57    if flag == 1 end
58         T=(rand(1,1)>0.5);
59    cla,clg;
60    kx=rand(1,1);
61    while kx==1;
62       kx=rand(1,1);
63    end
64    ky=rand(1,1);
65    while ky==1;
66       ky=rand(1,1);
67    end
68    [mm,SD,zp1]=mtar(Nsub,D,z,kx,ky);
69    zs1=zp1+(ss*sig*T);
70    subl1=scaling1(zs1,mm,SD,.01,R);
71    showr(subl1,.025,.05,.14,.3,(D-1)/2);
72    subl1=scaling1(zs1,mm,SD,1,R);
73    showr(subl1,.19,.05,.14,.3,(D-1)/2);
74    subl1=scaling1(zs1,mm,SD,1.5,R);
75    showr(subl1,.355,.05,.14,.3,(D-1)/2);
76    subl1=scaling1(zs1,mm,SD,2,R);
77    showr(subl1,.52,.05,.14,.3,(D-1)/2);
78    subl1=scaling1(zs1,mm,SD,2.5,R);
79    showr(subl1,.685,.05,.14,.3,(D-1)/2);
80    subl1=scaling1(zs1,mm,SD,3,R);
81    showr(subl1,.85,.05,.14,.3,(D-1)/2);
82
83    [mm1,SD1,zp2]=mtar(Nsub,D,z,kx,ky);
84    zs2=zp2+(ss*sig*T);
85    [subl2,scale]=scaling2(zs2,mm1,SD1,1.4,R);
```

```
86      showr(subl2,.025,.35,.14,.3,(D-1)/2);
87      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+1),R);
88      showr(subl2,.19,.35,.14,.3,(D-1)/2);
89      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+2),R);
90      showr(subl2,.355,.35,.14,.3,(D-1)/2);
91      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+3),R);
92      showr(subl2,.52,.35,.14,.3,(D-1)/2);
93      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+4),R);
94      showr(subl2,.685,.35,.14,.3,(D-1)/2);
95      [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+5),R);
96      showr(subl2,.85,.35,.14,.3,(D-1)/2);
97
98      [mm2,SD2,zp3]=mtar(Nsub,D,z,kx,ky);
99      zs3=zp3+(ss*sig*T);
100     [subl3,scale]=scaling3(zs3,mm2,SD2,1,R);
101     showr(subl3,.025,.65,.14,.3,(D-1)/2);
102     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+.5),R);
103     showr(subl3,.19,.65,.14,.3,(D-1)/2);
104     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+.75),R);
105     showr(subl3,.355,.65,.14,.3,(D-1)/2);
106     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
107     showr(subl3,.52,.65,.14,.3,(D-1)/2);
108     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1.5),R);
109     showr(subl3,.685,.65,.14,.3,(D-1)/2);
110     [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+2),R);
111     showr(subl3,.85,.65,.14,.3,(D-1)/2);
112
113     disp('target number is');
114     disp (count);
115     tgt = input('is there a target? [(y/n); ...
116               (y/n) exits test loop] >>','s');
117       if (tgt=='y'| tgt=='n')
118       count=count+1;
119       if T;
120          if tgt=='y'
121             disp('CORRECT');
122             c1=c1+1;
123          else
124             disp('WRONG');
125             c2=c2+1;
126          end
127       else
128          if tgt=='y'
129             disp('WRONG');
130             c4=c4+1;
131          else
132             disp('CORRECT');
133             c3=c3+1;
134          end
135       end
136       if count==floor(n/5) | count==floor(2*n/5) | count== ...
137             floor(3*n/5)  | count==floor(4*n/5) |count==n;
138          [PDO,PFAO,snrexp,flag]=check(c1,c2,c3,c4,snrnom,sigsnr);
139          save opwdata c1 c2 c3 c4 PDO PFAO snrexp sigsnr sig ...
140                       snrnom D Nsub;
141          writefil(count,c1,c2,c3,c4,snrexp,PDO,PFAO,sigsnr,sig, ...
142           .              snrnom,D,Nsub);
143       else
144       PDO=c1/(c1+c2);
145       PFAO=c4/(c3+c4);
146          if PDO<1 & PDO>0 & PFAO<1 &PFAO>0
147             t2sig=erfinv(1-2*PFAO);
148             snrexp=sqrt(2)*(t2sig-erfinv(1-2*PDO));
149             sprintf('PDO:%5.4f PFAO:%5.4f snrexp:%5.4f', ...
150                      PDO,PFAO,snrexp)
151          end
152       end
153       if flag==1 break; end
```

```
    else
       flag=1;
       break;
    end
end
PDO=(c1)/(c1+c2);
PFAO=(c4)/(c3+c4);
t2sig=erfinv(1-2*PFAO);
snrexp=sqrt(2)*(t2sig-erfinv(1-2*PDO));
```

OPMACH

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called opmach designed to run opwhite and machperf.
% The program obtains operator and machine probability of detection,
% probability of false alarm and graphs them for comparison.
%
%
%  INPUTS
% sig   (1,1)    = Input pixel value of target
% D   (1,1)      = Diameter of circular disc to be used in defining the
%                  filter; D must be odd.
% n     (1,1)    = number of displays
% Nsub  (1,num) = number of pixels on the x axis.
%
%
%  OUTPUTS
% PDO     (1,1)  = Probability the operator had a true detection
% PFAO    (1,1)  = Probability the operator thought there was a
%                  target but there was none
% PDMOS  (100,1)= Probability of true detection-report by optimum
%                  filter at operator signal
% PFAMOS(100,1) = Probability of false detection-report by optimum
%                  filter at operator  signal
%
%
%  [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig)


function [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig);
load start
[PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opnwhite(n,sig,z,D,Nsub);
[PDMOS,PFAMOS]=machperf(sig,D,z,PDO,PFAO,Nsub,ssow);
H=prbprb(PFA,PD);
prbprb(PFAO,PDO,H,'ro');
prbprb(PFAMOS,PDMOS,H,'g-');
load fnum;
fig=sprintf('exp#%1.0f',fnum)
```

PDGEN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called PDgen used to calculate a set of random error
% results for the estimated PDO, PFAO operator
% performance results.
%
%
%  INPUTS
%    n    (1,1)    = number of trials
% Nsub   (1,1)    = size of background matrix to display
% sig    (1,1)    = target signal level
% pill   (1,1)    = box containing the signal disc at the center
% D      (1,1)    = target diameter
% z    (256,256) = background to display
%
%  OUTPUTS
% sigsnr(1,1)     = standard deviation of random snr results
```

```
% snrbar(1,1)   = mean value of random snr results
% sig   (1,1)   = target signal level to conduct experiment with
%
% [PDO,PFAO]=PDgen(n,Nsub,D,sig,z,pill,M)

function [PDO,PFAO]=PDgen(n,Nsub,D,sig,z,pill)
[c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(n/5,Nsub,D,sig,z,pill);
sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
more = input('do you want to try again? (y/n); >>','s');
while more == 'y'
   nn=input('enter number of trials to conduct');
   sig2=input('enter the signal you want to use');
   [c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(nn,Nsub,D,sig2,z,pill);
   sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
   more = input('do you want to try again? (y/n); >>','s');
end
if more=='n'& PDO==1
   PDO=.999;
elseif more=='n'& PFAO==0
   PFAO=.001;
end
```

PDSNREST

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called pdsnrest used to calculate a set of random error
% results for the operator (PD,PFA)-pair.
%
%
%  INPUTS
% n    (1,1)  = number of trials
% Nsub (1,1)  = size of background matrix to display
% sig  (1,1)  = target signal level
% pill (1,1)  = box containing the signal disc at the center
% M    (1,1)  = number of monte carlo simulations to be run
% D    (1,1)  = target diameter
% z  (256,256)= background to display
%
%  OUTPUTS
% sigsnr(1,1) = standard deviation of random snr results
% snrbar(1,1) = mean value of random snr results
% sig (1,1)   = target signal level to conduct experiment with
%
% [sigsnr,snrbar,sig]=pdsnrest(n,Nsub,D,sig,z,pill,M)

function [sigsnr,snrbar,sig]=pdsnrest(n,Nsub,D,sig,z,pill,M);
if ~exist('M'); M=1e3; end
[c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(n/5,Nsub,D,sig,z,pill);
sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
more = input('do you want to try again? (y/n); >>','s');
while more == 'y'
   nn=input('enter number of trials to conduct');
   sig2=input('enter the signal you want to use');
   [c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(nn,Nsub,D,sig2,z,pill);
   sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
   more = input('do you want to try again? (y/n); >>','s');
end
if more=='n' & PDO<1 & PFAO>0
hN=n/2;
R=rand(1,hN*M);
ii=R<PDO;
ii=reshape(ii,hN,M);
TT=sum(ii);
R=rand(1,hN*M);
pnfanom=PDO;
ii=R<pnfanom;
ii=reshape(ii,hN,M);
```

```
NN=sum(ii);
ii=find(TT==hN|TT==0);
TT(ii)=[];
NN(ii)=[];
ii=find(NN==hN|NN==0);
TT(ii)=[];
NN(ii)=[];
PD=TT/hN;
PFA=1-NN/hN;
t2sig=erfinv(1-2*PFA);
s2sig=-erfinv(1-2*PD)+t2sig;
ransnr=s2sig*sqrt(2);
snrbar=mean(ransnr);
sigsnr=std(ransnr);
if exist('sig2') sig=sig2; end
A=100;
[Q,X]=hist(ransnr,A);
Q=Q(1:A-1);
X=X(1:A-1);
Q=cumsum(Q);
Q=Q/Q(length(Q));
linprb(Q,X);
figure(gcf);
end
```

PSDGEN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called psdgen used to develop a two dimensional power
% spectral density (psd) from a two dimensional background scene. Based
% on the assumption of isotropy a one dimensional result, psd versus
% spatial frequency magnitude, is developed. A single scene is used and
% ensemble averaging is "accomplished" by averaging over a ring in
% spatial frequency space. The program uses spline interpolation on the
% calculated set of psd values to generate a two dimensional array of
% psd values corresponding to the frequencies of the two dimensionnal
% FFT of the input background scene.
%
%
%  INPUTS
% N    (1,1)      = Number of spatial frequencies
% z    (256,256) = Background scene
%
%
%  OUTPUTS
% PSD (256,256) = 2-D version of psd
%
%    PSD=psdgen(N,z)

function PSD=psdgen(N,z);
zz=fft2(z);
k=0:127; k=[k -fliplr(k+1)];
k=ones(256,1)*k;
k=k.^2; k=k+k'; k=sqrt(k);
f=128*sqrt(2)*(1:N)/N;
df=(f(2)-f(1))/2;
psd=[];
for n=1:N
   ii= find(k>f(n)-df & k<=f(n)+df);
   psd = [psd sum((abs(zz(ii))).^2)/length(ii)];
end
psd(35)=(psd(34)+psd(36))/2;
k=k(:);
PSD=spline(f,psd,k);
PSD=reshape(PSD,256,256);

```

APPENDIX C SPOTC

This appendix contains a printout of the programs written which support experiment 3. This experiment uses SPOT background images. The control program is OPMACH. This program calls programs called OPCAM and MACHPERF. The OPCAM program collects the operator (P_D,P_{FA})-pair and the MACHPERF program collects the machine (P_D,P_{FA})-pairs.

The OPCAM program provides the operator with a training phase then when the operator feels comfortable with target recognition the program begins the experimental phase. The OPCAM program interfaces with the operator through the keyboard and the computer monitor. The computer monitor displays eighteen target/background images, and a decision box for the operator to input his choice of target present or not present. OPCAM calls the programs MAKEPILL, PDSNREST and EXOPR3. PDSNREST estimates the effective (P_D,P_{FA})-pair of an xperiment. PDSNREST allows the operator to have as many training trials as he requests. The program also allows the operator to adjust the target signature strength so that the effective (P_D,P_{FA})-pair is close to the nominal (P_D,P_{FA})-pair. The PDSNREST calls the program EXOPR3 which serves as the interface between the monitor and the operator and collects/calculates the operator effective (P_D,P_{FA})-pair. The EXOPR4 program serves as the interface between the operator and the computer monitor for the experimental data collection phase of the experiment. This program collects the effective (P_D,P_{FA})-pair collected as data. EXOPR4 calls the programs CHECK, WRITEFILE, CAMO, SHOW, SCALE(1,2,3), MTAR. MTAR adds a target of uniform signature to the background pixels. The MACHPERF program contains the algorithm to implement the machine/filter. The MACHPERF program in effect offers the same target/background definitions and parameters to the machine/filter as those shown to the operator. The MACHPERF program calls the programs PSDGEN, MAKEPILL, CAM. MAKEPILL and CAM serve the same purpose as in the operator collection phase. The program PSDGEN calculates the background matrix spatial frequency power spectral density (PSD) which is then used to implement the optimum linear filter.

EXPERIMENT 3

OPMACH

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called opmach designed to run opwhite and machperf.
% The program obtains operator and machine probability of detection,
% probability of false alarm and graphs them for comparison.
%
%
%  INPUTS
% sig    (1,1)      = Input pixel value of target
% D      (1,1)      = Diameter of circular disc to be used in defining
%                     the filter; D must be odd.
% n      (1,1)      = number of displays
% Nsub   (1,num)    = number of pixels on the x axis.
%
%
%  OUTPUTS
% PDO     (1,1)     = Probability the operator had a true detection
% PFAO    (1,1)     = Probability the operator thought there was a
%                     target but there was none
% PDMOS   (100,1)   = Probability of true detection-report by optimum
%                     filter at operator signal
% PFAMOS(100,1)     = Probability of false detection-report by optimum
%                     filter at operator signal
%
%
%  [PDO,PFAO,PFAMOS,PDMOS]=opmach(n,Nsub,D,sig)

```

```
27
28 function [PDO,PFAO,PFAMOS,PDMOS,sigm]=opmach(n,Nsub,D,sig);
29 load start;
30 [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opcam(n,sig,z,D,Nsub);
31 [PDMOS,PFAMOS,sigm]=machperf(sig,D,z,PDO,PFAO,Nsub,ssow);
32 H=prbprb(PFA,PD);
33 prbprb(PFAO,PDO,H,'ro');
34 prbprb(PFAMOS,PDMOS,H,'g*');
35 load fnum;
36 fig=sprintf('exp#%1.0f',fnum)
```

OPCAM

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 % Function called opcam designed to check the efffective
 3 % signal-to-noise (snreff) of an experiment. The function calculates an
 4 % operator,s snreff from the error function and the calculated
 5 % (PD,PFA)-pair. The program then compares this value with  nominal
 6 % signal-to noise-ratio (snrnom) of the to see if the operator is
 7 % within an acceptable limit (2 standard deviations).
 8 %
 9 %
10 %  INPUTS
11 % n     (1,1) =Total number of operator trials used in the experiment
12 % sig   (1,1) =Target signal level
13 % D     (1,1) =Target radius
14 % Nsub  (1,1) =Size of background matrix to display
15 % z  (256,256)=background to display
16 %
17 %
18 %  OUTPUTS
19 % snrexp(1,1) = Signal to noise ratio that the operator performs with
20 %     during the experiment. (snreff)
21 % sigsnr(1,1) = Standard deviation of the snrnom
22 % snrbar(1,1) = average snrnom
23 % PDO   (1,1) = operator probability of detection
24 % PFAO  (1,1) = operator probability of false alarm
25 %    c1 (2,1) = number of times target was present and
26 %               detected
27 %    c2 (2,1) = number of times target was present and not
28 %               detected
29 %    c3 (2,1) = number of times no target and no detection
30 %    c4 (2,1) = number of times no target and false detection
31 %
32 %
33 % [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]=opcam(n,sig,z,D,Nsub);
34
35
36
37 function [PDO,PFAO,snrbar,sigsnr,snrexp,c1,c2,c3,c4]= ...
38               opcam(n,sig,z,D,Nsub);
39 pill=makepill(Nsub,D);
40 [sigsnr,snrbar,sig]=pdsnrest(n,Nsub,D,sig,z,pill);
41 [c1,c2,c3,c4,flag,PDO,PFAO,snrexp]=exopr4(n,Nsub,D,sig,z, ...
42            snrbar,sigsnr,pill);
```

MACHPERF

```
 1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2 % Function called machperf designed to analyze performance in terms
 3 % of PD/PFA of a machine in looking for a uniform circular disc target
 4 % with imperfect camouflage. The target is present in a background
 5 % scene that is randomly chosen.The machine uses a set of filter
 6 % weights to recognize targets.
 7 %
 8 %
 9 %  INPUTS
```

```
% sig   (1,1)       = Input pixel value of target
% D    (1,1)        = Diameter of circular disc to be used in defining
%                     the filter; D must be odd.
% z    (256,256)    = Background scene
% PDO    (1,1)      = Operator probability of detection
% PFAO   (1,1)      = Operator probability of false alarm
% Nsub   (1,N)      = Size of background scene to display
% filt   (Nsub,Nsub)= filter weight array
%
%  OUTPUTS
% PDMOS   (100,1)   = Probability of true detection-report by optimum
%                     filter at operator signal
% PFAMOS(100,1)     = Probability of false detection-report by optimum
%                     filter at operator signal
% sigm   (1,1)      = signal machine needed to match the operator
%                     performance
%
%
%
%    [PDMOS,PFAMOS,sigm]=machperf(sig,D,z,PDO,PFAO,Nsub,filt);

function  [PDMOS,PFAMOS,sigm]=machperf(sig,D,z,PDO,PFAO,Nsub,filt);
sigm=sig;
R=(D-1)/2;
RN=(Nsub-1)/2;
x=(-RN:RN).^2;
x=ones(size(x'))*x;
R2=x+x';
ii=find(R2<=R^2);
L=length(ii);
if exist( 'ss');
   pill=makepill(Nsub,D);
   filt=pill;
else exist('ssow');
   pill=makepill(256,D);
   s=fft2(pill);
   PSD=psdgen(50,z);
   ssow=real(ifft2(s./PSD));
   P=(129-(Nsub-1)/2):(129+(Nsub-1)/2);
   ssow=ssow(P,P);
   filt=ssow;
end
pd=256-Nsub;
qd=256-Nsub;
Sum1=[];
N=1000;
X=[];
Y=[];
A=sum(filt(ii));
for b=1:N
   x= (RN+1)+pd*rand(1,1);
   y= (RN+1)+qd*rand(1,1);
   cam=camo(D,z,rand(1,1),rand(1,1));
   Z= z(x-RN:x+RN,y-RN:y+RN);
   Sum1=[Sum1; sum(sum(Z.*filt))];
   Z(ii)=ones(size(Z(ii)))*mean(Z(ii))+cam;
   Sum2=[Sum2; sum(sum(Z.*filt))];
end
Sums=[Sum1 Sum2+sigm*A];
T=linspace(min(min(Sums)),max(max(Sums)),102);
T(102)=[];
T(1)=[];
PDMOS=[];
PFAMOS=[];
for b=1:100
   e=(T(b) < Sums);
   e=sum(e);
   PDMOS=[PDMOS e(2)/N];
```

```
78      PFAMOS=[PFAMOS e(1)/N];
79  end
```

PSDGEN

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Function called psdgen used to develop a two dimensional power
3  spectral density  (psd) from a two dimensional background scene.
4  Based on the assumption of isotropy a one dimensional result, psd
5  versus spatial frequency magnitude, is developed. A single scene is
6  used and ensemble averaging is "accomplished" by averaging over a
7  ring in spatial frequency space.The program uses spline interpolation
8  on the calculated set of psd values to generate a two dimensional
9  array of psd values corresponding to the frequencies of the two
10 dimensionnal FFT of the input background scene.
11 %
12 %
13 %  INPUTS
14 % N    (1,1)      = Number of spatial frequencies
15 % z    (256,256) = Background scene
16 %
17 %
18 %  OUTPUTS
19 % PSD (256,256) = 2-D version of psd
20 %
21 %    PSD=psdgen(N,z)
22
23 function PSD=psdgen(N,z);
24 zz=fft2(z);
25 k=0:127; k=[k -fliplr(k+1)];
26 k=ones(256,1)*k;
27 k=k.^2; k=k+k'; k=sqrt(k);
28 f=128*sqrt(2)*(1:N)/N;
29 df=(f(2)-f(1))/2;
30 psd=[];
31 for n=1:N
32    ii= find(k>f(n)-df & k<=f(n)+df);
33    psd = [psd sum((abs(zz(ii))).^2)/length(ii)];
34 end
35 psd(35)=(psd(34)+psd(36))/2;
36 k=k(:);
37 PSD=spline(f,psd,k);
38 PSD=reshape(PSD,256,256);
```

EXOPR3

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % function called exopr3 used to collect pd/pfa related data. This
3  % function will display three series of target images. The first series
4  % displays a circular disc target in a background pattern. The second
5  % series displays a circular disc target in a background pattern where
6  % the background and target are between the minimum pixel value and the
7  % mean value of the pixels. The third series displays a circular disk
8  % target in a background pattern where the disc and the background
9  % pattern are between the mean background pixel value and the maximum
10 % background pixel value.  The monitor display is designed to provide
11 % the operator with the equivalent of a color/contrast control knob.
12 % The function collects operator performance numbers from which pd and
13 % pfa are calculated. The data collection is based on some specified
14 % number of separate displays. This program is designed for the
15 % pre-experiment stage when the operator is being trained and PD,PFA
16 % are being verified.
17 %
18 %              INPUTS
19 %    n         (1,1)       = number of displays
20 %   Nsub       (1,1)       = size of the background scene for display.
21 %    D         (1,1)       = circle Diameter
```

```
%   sig       (1,1)       = Target signal
%    z        (256,256)   = Random image field
%    ss       (Nsub,Nsub)= Box containing signal disc at center
%
%
%             OUTPUTS
%    c1       (2,1)       = number of times target was present and
%                           detected
%    c2       (2,1)       = number of times target was present and not
%                           detected
%    c3       (2,1)       = number of times no target and no detection
%    c4       (2,1)       = number of times no target and false detection
%    flag     (1,1)       = emergency exit indicator,
%                           (1/0)=(e-exit/norm-end)
%    PD       (1,1)       = Probability the target was present and the
%                           operator identified it
%    PFA      (1,1)       = Probability the target was not present but
%                           the operator thought it was present
%
%
%   [c1,c2,c3,c4,flag,PD,PFA]=exopr3(n,Nsub,D,sig,z,ss)


function [c1,c2,c3,c4,flag,PD,PFA]=exopr3(n,Nsub,D,sig,z,ss);
if n>25 n=25; end;
R=(D-1)/2;
c1=0;c2=0;c3=0;c4=0;
cla,clg,axis('off');
flag=0;
count=1;
for k=1:n
        if flag == 1 end
        T=(rand(1,1)>0.5);
        cla,clg;
        kx=rand(1,1);
        while kx==1;
                kx=rand(1,1);
        end
        ky=rand(1,1);
        while ky==1;
                ky=rand(1,1);
        end
        if T cam=camo(D,z,kx,ky);end
        jj=find(ss==1);
        [mm,SD,zp1]=mtar(Nsub,D,z,kx,ky);
        if T zp1(jj)=mean(zp1(jj))+cam; end;
        zs1=zp1+(ss*sig*T);
        subl1=scaling1(zs1,mm,SD,.01,R);
        showr(subl1,.025,.05,.14,.3,(D-1)/2);
        subl1=scaling1(zs1,mm,SD,1,R);
        showr(subl1,.19,.05,.14,.3,(D-1)/2);
        subl1=scaling1(zs1,mm,SD,1.5,R);
        showr(subl1,.355,.05,.14,.3,(D-1)/2);
        subl1=scaling1(zs1,mm,SD,2,R);
        showr(subl1,.52,.05,.14,.3,(D-1)/2);
        subl1=scaling1(zs1,mm,SD,2.5,R);
        showr(subl1,.685,.05,.14,.3,(D-1)/2);
        subl1=scaling1(zs1,mm,SD,3,R);
        showr(subl1,.85,.05,.14,.3,(D-1)/2);

        [mm1,SD1,zp2]=mtar(Nsub,D,z,kx,ky);
        if T zp2(jj)=mean(zp2(jj))+cam; end;
        zs2=zp2+(ss*sig*T);
        [subl2,scale]=scaling2(zs2,mm1,SD1,1.4,R);
        showr(subl2,.025,.35,.14,.3,(D-1)/2);
        [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+1),R);
        showr(subl2,.19,.35,.14,.3,(D-1)/2);
```

```
        [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+2),R);
        showr(subl2,.355,.35,.14,.3,(D-1)/2);
        [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+3),R);
        showr(subl2,.52,.35,.14,.3,(D-1)/2);
        [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+4),R);
        showr(subl2,.685,.35,.14,.3,(D-1)/2);
        [subl2,scale]=scaling2(zs2,mm1,SD1,(scale+5),R);
        showr(subl2,.85,.35,.14,.3,(D-1)/2);

        [mm2,SD2,zp3]=mtar(Nsub,D,z,kx,ky);
        if T zp3(jj)=mean(zp3(jj))+cam; end;
        zs3=zp3+(ss*sig*T);
        [subl3,scale]=scaling3(zs3,mm2,SD2,.5,R);
        showr(subl3,.025,.65,.14,.3,(D-1)/2);
        [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
        showr(subl3,.19,.65,.14,.3,(D-1)/2);
        [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
        showr(subl3,.355,.65,.14,.3,(D-1)/2);
        [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
        showr(subl3,.52,.65,.14,.3,(D-1)/2);
        [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
        showr(subl3,.685,.65,.14,.3,(D-1)/2);
        [subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
        showr(subl3,.85,.65,.14,.3,(D-1)/2);



        disp('target number is');
        disp (count);
        tgt = input('is there a target? [(y/n); ~(y/n) exits test loop]
                ...  >>','s');
        if (tgt=='y'| tgt=='n')
        count=count+1;
                if T;
                        if tgt=='y'
                                disp('CORRECT');
                                c1=c1+1;
                        else
                                disp('WRONG');
                                c2=c2+1;
                        end
                else
                        if tgt=='y'
                                disp('WRONG');
                                c4=c4+1;
                        else
                                disp('CORRECT');
                                c3=c3+1;
                        end
                end
        else
                flag=1;
                break;
        end
end
PD=(c1)/(c1+c2);
PFA=(c4)/(c3+c4);
```

EXOPR4

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called exopr4 used to collect pd/pfa related data. This
% function will display three series of target images. The first series
% displays a circular disc target in a background pattern. The second
% series displays a circular disc target in a background pattern where
% the background and target are between the minimum pixel value and the
```

```
% mean value of the pixels. The third series displays a circular disk
% target in a background pattern where the disc and the background
% pattern are between the maximum background pixel value and the
% maximum background pixel value.  The monitor display is meant to
% provide the operator with the equivalent of a color contrast control
% knob. The function collects operator performance numbers from which
% pd and pfa are calculated.The data collection is based on some
% specified number of separate displays. The collects the operator
% (PD,PFA)-pair for an for an experiment.
%              INPUTS
%    n         (1,1)       = number of displays
%   Nsub       (1,1)       = size of the background scene for display.
%    D         (1,1)       = circle Diameter
%   sig        (1,1)       = Target signal
%    z        (256,256)    = Background scene to display
%  snrnom      (1,1)       = nominal signal to noise voltage ratio
%  sigsnr      (1,1)       = standard deviation of the
%                            signal-to-noise ratio
%    ss      (Nsub,Nsub)   = Box containing signal disc at center
%
%
%              OUTPUTS
%    c1        (2,1)       = number of times target was present and
%                            detected
%    c2        (2,1)       = number of times target was present and not
%                            detected
%    c3        (2,1)       = number of times no target and no detection
%    c4        (2,1)       = number of times no target and false
%                            detection
%    flag      (1,1)       = emergency exit indicator,
%                            (1/0)=(e-exit/norm-end)
%    PD0       (1,1)       = Probability the target was present and the
%                            operator identified it
%    PFA0      (1,1)       = Probability the target was not present but
%                            the operator thought it was present
% snrexp       (1,1)       = Signal to noise voltage ratio of the
%                            operator  during the experiment
%
%
% [c1,c2,c3,c4,flag,PD0,PFA0,snrexp]=exopr4(n,Nsub,D,sig,z, ...
%       snrnom,sigsnr,ss)

function [c1,c2,c3,c4,flag,PD0,PFA0,snrexp]=exopr4(n,Nsub,D,sig,z, ...
        snrnom,sigsnr,ss);
R=(D-1)/2;
c1=0;c2=0;c3=0;c4=0;
cla,clg,axis('off');
flag=0;
count=1;
for k=1:n
   if flag == 1 end
        T=(rand(1,1)>0.5);
   cla,clg;
   kx=rand(1,1);
   while kx==1;
      kx=rand(1,1);
   end
   ky=rand(1,1);
   while ky==1;
      ky=rand(1,1);
   end
   if T cam=camo(D,z,kx,ky);end
   jj=find(ss==1);
   [mm,SD,zp1]=mtar(Nsub,D,z,kx,ky);
   if T zp1(jj)=mean(zp1(jj))+cam;end
   zs1=zp1+(ss*sig*T);
   subl1=scaling1(zs1,mm,SD,.01,R);
   showr(subl1,.025,.05,.14,.3,(D-1)/2);
```

```
subl1=scaling1(zs1,mm,SD,1,R);
showr(subl1,.19,.05,.14,.3,(D-1)/2);
subl1=scaling1(zs1,mm,SD,1.5,R);
showr(subl1,.355,.05,.14,.3,(D-1)/2);
subl1=scaling1(zs1,mm,SD,2,R);
showr(subl1,.52,.05,.14,.3,(D-1)/2);
subl1=scaling1(zs1,mm,SD,2.5,R);
showr(subl1,.685,.05,.14,.3,(D-1)/2);
subl1=scaling1(zs1,mm,SD,3,R);
showr(subl1,.85,.05,.14,.3,(D-1)/2);

[mm1,SD1,zp2]=mtar(Nsub,D,z,kx,ky);
if T zp2(jj)=mean(zp2(jj))+cam;end
zs2=zp2+(ss*sig*T);
[subl2,scale]=scaling2(zs2,mm1,SD1,1.4,R);
showr(subl2,.025,.35,.14,.3,(D-1)/2);
[subl2,scale]=scaling2(zs2,mm1,SD1,(scale+1),R);
showr(subl2,.19,.35,.14,.3,(D-1)/2);
[subl2,scale]=scaling2(zs2,mm1,SD1,(scale+2),R);
showr(subl2,.355,.35,.14,.3,(D-1)/2);
[subl2,scale]=scaling2(zs2,mm1,SD1,(scale+3),R);
showr(subl2,.52,.35,.14,.3,(D-1)/2);
[subl2,scale]=scaling2(zs2,mm1,SD1,(scale+4),R);
showr(subl2,.685,.35,.14,.3,(D-1)/2);
[subl2,scale]=scaling2(zs2,mm1,SD1,(scale+5),R);
showr(subl2,.85,.35,.14,.3,(D-1)/2);

[mm2,SD2,zp3]=mtar(Nsub,D,z,kx,ky);
if T zp3(jj)=mean(zp3(jj))+cam;end
zs3=zp3+(ss*sig*T);
[subl3,scale]=scaling3(zs3,mm2,SD2,1,R);
showr(subl3,.025,.65,.14,.3,(D-1)/2);
[subl3,scale]=scaling3(zs3,mm2,SD2,(scale+.5),R);
showr(subl3,.19,.65,.14,.3,(D-1)/2);
[subl3,scale]=scaling3(zs3,mm2,SD2,(scale+.75),R);
showr(subl3,.355,.65,.14,.3,(D-1)/2);
[subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1),R);
showr(subl3,.52,.65,.14,.3,(D-1)/2);
[subl3,scale]=scaling3(zs3,mm2,SD2,(scale+1.5),R);
showr(subl3,.685,.65,.14,.3,(D-1)/2);
[subl3,scale]=scaling3(zs3,mm2,SD2,(scale+2),R);
showr(subl3,.85,.65,.14,.3,(D-1)/2);

disp('target number is');
disp (count);
tgt = input('is there a target? [(y/n); ~(y/n) exits ...
            test loop] >>','s');
     if (tgt=='y'| tgt=='n')
   count=count+1;
   if T;
      if tgt=='y'
         disp('CORRECT');
         c1=c1+1;
      else
         disp('WRONG');
         c2=c2+1;
      end
   else
      if tgt=='y'
         disp('WRONG');
         c4=c4+1;
      else
         disp('CORRECT');
         c3=c3+1;
      end
   end
   if count==floor(n/5) | count==floor(2*n/5) | count== ...
         floor(3*n/5)  | count==floor(4*n/5) |count==n;
```

```
143           [PDO,PFAO,snrexp,flag]=check(c1,c2,c3,c4,snrnom,sigsnr);
144           save opwdata c1 c2 c3 c4 PDO PFAO snrexp sigsnr sig ...
145              snrnom D Nsub;
146           writefil(count,c1,c2,c3,c4,snrexp,PDO,PFAO,sigsnr,sig, ...
147              snrnom,D,Nsub);
148        else
149        PDO=c1/(c1+c2);
150        PFAO=c4/(c3+c4);
151           if PDO<1 & PDO>0 & PFAO<1 &PFAO>0
152              t2sig=erfinv(1-2*PFAO);
153              snrexp=sqrt(2)*(t2sig-erfinv(1-2*PDO));
154              sprintf('PDO:%5.4f PFAO:%5.4f snrexp:%5.4f',PDO, ...
155                 PFAO,snrexp)
156           end
157        end
158        if flag==1 break; end
159     else
160        flag=1;
161        break;
162     end
163  end
164  PDO=(c1)/(c1+c2);
165  PFAO=(c4)/(c3+c4);
166  t2sig=erfinv(1-2*PFAO);
167  snrexp=sqrt(2)*(t2sig-erfinv(1-2*PDO));
```

PDGEN

```
  1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  2  % Function called PDgen used to calculate a set of random error
  3  % results for the estimated PDO, PFAO operator
  4  % performance results.
  5  %
  6  %
  7  %  INPUTS
  8  % n     (1,1)   = number of trials
  9  % Nsub  (1,1)   = size of background matrix to display
 10  % sig   (1,1)   = target signal level
 11  % pill  (1,1)   = box containing the signal disc at the center
 12  % D     (1,1)   = target diameter
 13  % z   (256,256) = background to display
 14  %
 15  %  OUTPUTS
 16  % sigsnr(1,1)   = standard deviation of random snr results
 17  % snrbar(1,1)   = mean value of random snr results
 18  % sig   (1,1)   = target signal level to conduct experiment with
 19  %
 20  % [PDO,PFAO]=PDgen(n,Nsub,D,sig,z,pill,M)
 21
 22  function [PDO,PFAO]=PDgen(n,Nsub,D,sig,z,pill)
 23  [c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(n/5,Nsub,D,sig,z,pill);
 24  sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
 25  more = input('do you want to try again? (y/n); >>','s');
 26  while more == 'y'
 27     nn=input('enter number of trials to conduct');
 28     sig2=input('enter the signal you want to use');
 29     [c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(nn,Nsub,D,sig2,z,pill);
 30     sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
 31     more = input('do you want to try again? (y/n); >>','s');
 32  end
 33  if more=='n'& PDO==1
 34     PDO=.999;
 35  elseif more=='n'& PFAO==0
 36     PFAO=.001;
 37  end
```

PDSNREST

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called pdsnrest used to calculate a set of random error
% results for the estimated operator (PD,PFA)-pair.  The program first
% runs a set number of trials then calculates the operator
% experimental (PD,PFA)-pair. The program displays the experimental
% (PD,PFA)-pair to the operator and allows the operator to adjust the
% experimental signal. When the operator is satisfied with the
% experimental (PD,PFA)-pair the program use this experimental
% (PD,PFA)-pair to calculate the effective signal-to-noise ratio
% (called snrbar in this program) and the standard deviation (sigsnr)
% of snrbar.
%
%  INPUTS
% n      (1,1)    = number of trials
% Nsub   (1,1)    = size of background matrix to display
% sig    (1,1)    = target signal level
% pill   (1,1)    = box containing the signal disc at the center
% M      (1,1)    = number of monte carlo simulations to be run
% D      (1,1)    = target diameter
% z    (256,256)  = background to display
%
%  OUTPUTS
% sigsnr(1,1)     = standard deviation of random snr results
% snrbar(1,1)     = mean value of random snr results.
% sig    (1,1)    = target signal level to conduct experiment with
%
% [sigsnr,snrbar,sig]=pdsnrest(n,Nsub,D,sig,z,pill,M)

function [sigsnr,snrbar,sig]=pdsnrest(n,Nsub,D,sig,z,pill,M);
if ~exist('M'); M=1e3; end
[c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(n/5,Nsub,D,sig,z,pill);
sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
more = input('do you want to try again? (y/n); >>','s');
while more == 'y'
   nn=input('enter number of trials to conduct');
   sig2=input('enter the signal you want to use');
   [c1,c2,c3,c4,flag,PDO,PFAO]=exopr3(nn,Nsub,D,sig2,z,pill);
   sprintf('PDO is :%5.4f PFAO is:%5.4f',PDO,PFAO)
   more = input('do you want to try again? (y/n); >>','s');
end
if more=='n' & PDO<1 & PFAO>0
   hN=n/2;
   R=rand(1,hN*M);
   ii=R<PDO;
   ii=reshape(ii,hN,M);
   TT=sum(ii);
   R=rand(1,hN*M);
   pnfanom=PDO;
   ii=R<pnfanom;
   ii=reshape(ii,hN,M);
   NN=sum(ii);
   ii=find(TT==hN|TT==0);
   TT(ii)=[];
   NN(ii)=[];
   ii=find(NN==hN|NN==0);
   TT(ii)=[];
   NN(ii)=[];
   PD=TT/hN;
   PFA=1-NN/hN;
   t2sig=erfinv(1-2*PFA);
   s2sig=-erfinv(1-2*PD)+t2sig;
   ransnr=s2sig*sqrt(2);
   snrbar=mean(ransnr);
   sigsnr=std(ransnr);
   if exist('sig2') sig=sig2; end
   A=100;
```

```
    [Q,X]=hist(ransnr,A);
    Q=Q(1:A-1);
    X=X(1:A-1);
    Q=cumsum(Q);
    Q=Q/Q(length(Q));
    linprb(Q,X);
    figure(gcf);
end
```

PSDGEN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called psdgen used to develop a two dimensional power
spectral density  (psd) from a two dimensional background scene.
Based on the assumption of isotropy a one dimensional result, psd
versus spatial frequency magnitude, is developed. A single scene is
used and ensemble averaging is "accomplished" by averaging over a
ring in spatial frequency space.The program uses spline interpolation
on the calculated set of psd values to generate a two dimensional
array of psd values corresponding to the frequencies of the two
dimensionnal FFT of the input background scene.
%
%
%  INPUTS
% N   (1,1)      = Number of spatial frequencies
% z   (256,256) = Background scene
%
%
%  OUTPUTS
% PSD (256,256) = 2-D version of psd
%
%    PSD=psdgen(N,z)

function PSD=psdgen(N,z);
zz=fft2(z);
k=0:127; k=[k -fliplr(k+1)];
k=ones(256,1)*k;
k=k.^2; k=k+k'; k=sqrt(k);
f=128*sqrt(2)*(1:N)/N;
df=(f(2)-f(1))/2;
psd=[];
for n=1:N
   ii= find(k>f(n)-df & k<=f(n)+df);
   psd = [psd sum((abs(zz(ii))).^2)/length(ii)];
end
psd(35)=(psd(34)+psd(36))/2;
k=k(:);
PSD=spline(f,psd,k);
PSD=reshape(PSD,256,256);
```

CAMO

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Function called camo. Used to generate a zero mean camouflaged
%  disk pattern. The pattern corresponds to a randomly selected
%  portion of the scene.
%
%
%       INPUTS
%   D     (1,1)      = Diameter of circle region.
%   z     (M,N)      = Random image field.
%   ky    (1,1)      = Random number (uniform,zero to one)
%   kx    (1,1)      = Random number (uniform,zero to one)
%
%
%   OUTPUTS
%  cam   (1,ii)      = Randomly chosen piece of background matrix.
```

```
16 %
17 %
18 %
19 %        cam=camo(D,z,kx,ky);
20
21
22 function cam=camo(D,z,kx,ky);
23 [M,N]=size(z);
24 R=(D-1)/2;
25 x=(-R:R).^2;
26 x=ones(size(x'))*x;
27 R2=x+x';
28 ii=find(R2<=R^2);
29 L=length(ii);
30 m=1+R+floor((M-(D-1))*kx);
31 n=1+R+floor((N-(D-1))*ky);
32 cam=z(m-R:m+R,n-R:n+R);
33 mcirc=sum(cam(ii))/L;
34 cam=cam(ii)-mcirc;
```

APPENDIX C COMMON PROGRAMS

This appendix contains the text for the programs which are shared by all three experiments. The program GLINES plots the lines on the probability graph which correspond to a signal-to-noise ratios of (1-10). The program GAUSS is used to convert probabilities to the appropriate value corresponding to their location on the prbprb graph. The program converts probability distributions to the corresponding number of standard deviations of a gaussian distribution. The program called MAXMIN is used to find the appropriate axes scales to accomodate the entire range of the probability distribution to be plotted. The program PDPFAEFF calculates the operator effective (P_D,P_{FA})-pair using "Monte Carlo" techniques. The program PRBPRBER plots a (P_D,P_{FA})-pair on a prbprb graph with error bars. The program PRPRLAER plots (P_D,P_{FA})-pairs on a prbprb graph with labels on the effective signal-to-noise ratio lines. The program SNRCALC calculates the signal to noise ratio in gaussian noise of a given (P_D,P_{FA})-pair assuming an unbiased threshold. SNRESTER estimates the effective snr of an experiment and it's standard deviation based on "Monte Carlo" techniques and the number of experimental trials. MAKEPILL outputs the matched filter of a disc shaped target. 3 programs (SCALE1, SCALE2, SCALE3) each scale and limit the target/background images prior to display. Each scale program is used with different limits to display six background/target images. The three programs in effect serve as the operator's color/contrast control knob. SHOW displays the eighteen background/target images on the computer monitor in greyscale format. WRITEFILE stores the experimental data to both a text and a data file.The program CHECK compares the effective signal-to-noise ratio of an experiment with the nominal signal-to-noise ratio and allows the operator to terminate the experiment.

COMMON PROGRAMS

GLINES

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called glines used to generate PD PFA data sets for various
% signal to noise ratios. The noise is taken to be zero mean gaussian.
%
%  INPUTS
% SNR (1,N) =  set of signal to noise ratios
%
%
%  OUTPUTS
% PFA(400,1)= set of false alarm probabilities for threshold to noise
%             ratios from minus ten to plus ten
% PD (400,N)= set of detection probabilities for the same thresholds,
%             with each column corresponding to one of the SNR values.
%
%    [PFA,PD] = glines(SNR)

function [PFA,PD]= glines(SNR)
T=linspace(-10,10,400);
PFA=0.5*(1-erf(T'/sqrt(2)));
PD=[];
for s=SNR
   PD=[PD 0.5*(1-erf((T-s)'/sqrt(2)))];
end
```

GAUS

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% gaus. Function used to convert probability distributions to the
% corresponding number of standard deviations of a gaussian
% distribution.
%
```

```
%  INPUTS
% P    (m,n) =  matrix of probabilities
%  OUTPUTS
% sds (m,n) =  corresponding matrix of standard deviations
%
% sds=gaus(p)
%

function sds= gaus(p)
sds=sqrt(2)*erfinv(2*p-1);
ii=find(sds==-inf | sds==inf);
sds(ii)=NaN*ones(size(ii));
```

MAXMIN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called maxmin used to find maximum and minimum values in
% each column of an array. The max and min values are explicitly to
% ignore the presence ofNaN.
%
%  INPUTS
% A    (M,N) =  Array whose max values are to be found
%
%  OUTPUTS
% mx   (1,N) =  String of maximum values
% mn   (1,N) =  String of minimum values
%
%  [mx,mn]=maxmin(A);

function [mx,mn]=maxmin(A)
[M,N]=size(A);
mx=[]; mn=[];
for n=1:N
   a=A(:,n);
   ii=find(isnan(a));
   a(ii)=zeros(size(ii));
   b=max(a);
   if b==0
      b=min(a);
      a(ii)=b*ones(size(ii));
      b=max(a);
   end
   mx=[mx b];
   b=min(a);
   if b==0
      b=max(a);
      a(ii)=b*ones(size(ii));
      b=min(a);
   end
   mn=[mn b];
end
```

PDPFAEFF

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called PDPFAeff used to calculate a set of random error
% results for the estimated (PD,PFA)-pair for operator
% (random) performance results. The PD, PFA random performance results
% are drawn from a set whose distribution is governed by some nominal
% snr and the number of trials. It is assumed that half the trials
% have targets and half do not, and that the operator's effective
% threshold is at half the nominal snr.
%
%
%  INPUTS
% N      (1,1)  = number of trials
% snrnom(1,1)   = nominal signal to noise ratio
```

```
% M     (1,1)  = number of monte carlo simulations to be run
%
%
%  OUTPUTS
% sigPDc (1,1) = standard deviation of random PD results using
%                std(PD).
% sigPFAc(1,1) = standard deviation of random PD results " " ".
% PDM    (1,1) = Monte Carlo Probability of detection.
% PFAM   (1,1) = Monte Carlo probability of false alarm.
%
%
% [sigPDc,sigPFAc,PDM,PFAM]=PDPFAeff(N,snrnom,M)

function [sigPDc,sigPFAc,PDM,PFAM]=PDPFAeff(N,snrnom,M)
if ~exist('M'); M=1e3; end
p=(1-erf(snrnom/(2*sqrt(2))))/2;
q=1-p;
hN=N/2;
R=rand(1,hN*M);
ii=R<q;
ii=reshape(ii,hN,M);
TT=sum(ii);
ii=R<q;
ii=reshape(ii,hN,M);
NN=sum(ii);
ii=find(TT==hN|TT==0);
TT(ii)=[];
NN(ii)=[];
ii=find(NN==hN|NN==0);
TT(ii)=[];
NN(ii)=[];
PD=TT/hN;
PFA=1-NN/hN;
PDM=mean(PD);
PFAM=mean(PFA);
sigPDc=std(PD);
sigPFAc=std(PFA);
```

PRBPRBER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% prbprber Function used to plot a pair of probability
% distributions against each other on a graph whose scales each
% correspond to the gaussian distribution . This program displays
% error bars on operator data point. This program allows the user
% to select the axis size of the graph.
%
%
%  INPUTS
% P1  (1 or m,n)   = Set of probabilities associated with horizontal
%                    axis by convention PFA
% P2  (m,n)        = Set of probabilities associated with vertical axis
%                    by convention PD
% h   (1,1)        = handle for graph on which P1,P2 are to be plotted
% s   (string)     = string indicating how data is to be displayed
%
%  OUTPUTS
% H   (1,1)        = Handle of graph on which P1, P2 have been plotted
%
%
%
% H=prbprber(P1,P2,h,s)
%
function H=prbprber(P1,P2,h,s)
if exist('h')
   if isstr(h); s=h; clear h; end
end
```

```
format long E
lab1 = ['       ';'  1e-9 ';'       ';'  1e-7 ';'       ';'  1e-5 '; ...
      '       '; '0.001  ';'0.01   ';'0.1    ';'0.2    ';'       '; ...
      '       ';'0.5    ';'       ';'       ';'0.8    ';'0.9    '; ...
      '0.99   ';'0.999  ';' 4-9s  ';' 5-9s  ';'       ';' 7-9s  '; ...
'       ';' 9-9s  ';'       '];
labels = [1e-10 1e-9 1e-8 1e-7 1e-6 1e-5 0.0001 0.001  0.01   0.1   ...
   0.2    0.3     0.4    0.5    0.6    0.7000 0.8000 0.9000 0.9900 ...
 0.9990 1-1e-4 1-1e-5 1-1e-6 1-1e-7 1-1e-8 1-1e-9 1-1e-10];
[jj,ii]=maxmin(gaus(P1)); jj=max(jj); ii=min(ii);
ii=find(gaus(labels)<ii); ii=max(ii); if ii==[]; ii=1; end
jj=find(gaus(labels)>jj); jj=min(jj); if jj==[]; jj=27; end
lab1x=lab1(ii:jj,1:7); labelsx=labels(ii:jj);
[jj,ii]=maxmin(gaus(P2)); jj=max(jj); ii=min(ii);
ii=find(gaus(labels)<ii); ii=max(ii); if ii==[]; ii=1; end
jj=find(gaus(labels)>jj); jj=min(jj); if jj==[]; jj=27; end
lab1y=lab1(ii:jj,1:7); labelsy=labels(ii:jj);
labelsx=gaus(labelsx);
labelsy=gaus(labelsy);
if ~exist('h')
   hold off
   cla reset,clf
   delete(H)
   H=gca
else
   H=h;
end
set(H,'YLim',[min(labelsy) max(labelsy)]);
set(H,'XLim',[min(labelsx) max(labelsx)]);
set(H,'FontSize',6);
set(H,'xtick',labelsx,'xticklabels',lab1x,'box','on');
set(H,'ytick',labelsy,'yticklabels',lab1y);
hold on
N=input('How many trials did you run in the experiment?  ')
if exist('s')
   t2sig=erfinv(1-2*P1);
   s2sig=-erfinv(1-2*P2)+t2sig;
   snrnom=s2sig*sqrt(2);
   [sigPDc,sigPFAc,PDm,PFAm]=PDPFAeff(N,snrnom)
   PFAe=[P1-sigPFAc P1 P1+sigPFAc];
   PDe=[P2-sigPDc P2 P2+sigPDc];
   sdv=gaus(P2);sdh=gaus(P1);
   plot(sdh,sdv,'*')
   sdve=gaus(PDe);sdhe=gaus(PFAe);
   plot(sdhe,sdv*ones(size(sdhe)));
   plot(sdh*ones(size(sdve)),sdve);

else
   t2sig=erfinv(1-2*P1);
   s2sig=-erfinv(1-2*P2)+t2sig;
   snrnom=s2sig*sqrt(2);
   [sigPDc,sigPFAc,PDm,PFAm]=PDPFAeff(N,snrnom)
   PFAe=[P1-sigPFAc P1 P1+sigPFAc];
   PDe=[P2-sigPDc P2 P2+sigPDc];
   sdv=gaus(P2);sdh=gaus(P1);
   plot(sdh,sdv)
   sdve=gaus(PDe);sdhe=gaus(PFAe);
   plot(sdhe,sdv*ones(size(sdhe)));
   plot(sdh*ones(size(sdve)),sdve);

   end
xlabel('PFA');
ylabel('PD');
fig
size=input(' Do you want to size the graphics box ? y/n ', 's');
if size=='y'
   ax=input(' What size axis do you want 1e-1-1e-10');
   prbprb([ax (1-ax)],[(1-ax) ax],H,'.')
```

```
else
end
```

PRPRLAER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% prprlaer Function used to plot a pair of probability
% distributions against each other on a graph whose scales each
% correspond to the gaussian distribution and have gaussline labels
% and error bars on data.
%
%  INPUTS
% P1  (1 or m,n)= Set of probabilities associated with horizontal axis
%                 by convention PFA
% P2  (m,n)     = Set of probabilities associated with vertical axis
%                 by convention PD
% h   (1,1)     = handle for graph on which P1,P2 are to be plotted
% s   (string)  = string indicating how data is to be displayed
%
%  OUTPUTS
% H   (1,1)     = Handle of graph on which P1, P2 have been plotted
%
%
%
% H=prprlaer(P1,P2,h,s)
%
function H=prprlaer(P1,P2,h,s)
if exist('h')
   if isstr(h); s=h; clear h; end
end
format long E
snrlabh=1e-9;
snrlabv=[1e-6 0.00007 0.005 0.05 0.3 0.6 0.9 0.99 0.9995 1-1e-5];
lab1 = ['       ';'  1e-9 ';'       ';'  1e-7 ';'       '; ...
'  1e-5 ';'       '; '0.001  ';'0.01   ';'0.1    '; ...
'0.2    ';'       ';'       ';'0.5    ';'       '; ...
'       ';'0.8    ';'0.9    ';'0.99   ';'0.999  ';' 4-9s  '; ...
' 5-9s  ';'       ';' 7-9s  ';'       ';' 9-9s  ';'       '];
labels = [1e-10 1e-9 1e-8 1e-7 1e-6 1e-5 0.0001 0.001  0.01   0.1 ...
      0.2    0.3    0.4    0.5    0.6    0.7000 0.8000 0.9000 0.9900 ...
 0.9990 1-1e-4 1-1e-5 1-1e-6 1-1e-7 1-1e-8 1-1e-9 1-1e-10];
[jj,ii]=maxmin(gaus(P1)); jj=max(jj); ii=min(ii);
ii=find(gaus(labels)<ii); ii=max(ii); if ii==[]; ii=1; end
jj=find(gaus(labels)>jj); jj=min(jj); if jj==[]; jj=27; end
lab1x=lab1(ii:jj,1:7); labelsx=labels(ii:jj);
[jj,ii]=maxmin(gaus(P2)); jj=max(jj); ii=min(ii);
ii=find(gaus(labels)<ii); ii=max(ii); if ii==[]; ii=1; end
jj=find(gaus(labels)>jj); jj=min(jj); if jj==[]; jj=27; end
lab1y=lab1(ii:jj,1:7); labelsy=labels(ii:jj);
labelsx=gaus(labelsx);
labelsy=gaus(labelsy);
snrlabh=gaus(snrlabh);
snrlabv=gaus(snrlabv);
if ~exist('h')
   hold off
   cla reset,clf
   delete(H)
   H=gca
else
   H=h;
end
set(H,'YLim',[min(labelsy) max(labelsy)]);
set(H,'XLim',[min(labelsx) max(labelsx)]);
set(H,'FontSize',6);
set(H,'xtick',labelsx,'xticklabels',lab1x,'box','on');
set(H,'ytick',labelsy,'yticklabels',lab1y);
for T=1:10
```

```
    if T==1
       Glab=sprintf('snr= %1.0f',T);
       text(snrlabh,snrlabv(T),Glab);
    else
       Glab=sprintf('%1.0f',T);
       text(snrlabh,snrlabv(T),Glab);
    end
end

hold on
N=input('How many trials did you run in the experiment?  ');
if exist('s')
    t2sig=erfinv(1-2*P1);
    s2sig=-erfinv(1-2*P2)+t2sig;
    snrnom=s2sig*sqrt(2);
    [sigPDc,sigPFAc,PDm,PFAm]=PDPFAeff(N,snrnom)
    PFAe=[P1-sigPFAc P1 P1+sigPFAc];
    PDe=[P2-sigPDc P2 P2+sigPDc];
    sdv=gaus(P2);sdh=gaus(P1);
    plot(sdh,sdv,'*')
    sdve=gaus(PDe);sdhe=gaus(PFAe);
    plot(sdhe,sdv*ones(size(sdhe)));
    plot(sdh*ones(size(sdve)),sdve);
else
    t2sig=erfinv(1-2*P1);
    s2sig=-erfinv(1-2*P2)+t2sig;
    snrnom=s2sig*sqrt(2);
    [sigPDc,sigPFAc,PDm,PFAm]=PDPFAeff(N,snrnom)
    PFAe=[P1-sigPFAc P1 P1+sigPFAc];
    PDe=[P2-sigPDc P2 P2+sigPDc];
    sdv=gaus(P2);sdh=gaus(P1);
    plot(sdh,sdv)
    sdve=gaus(PDe);sdhe=gaus(PFAe);
    plot(sdhe,sdv*ones(size(sdhe)));
    plot(sdh*ones(size(sdve)),sdve);
end
xlabel('PFA');
ylabel('PD');
prbprb([.0000000001 .9999999999],[.9999999999 .0000000001],H,'.');
hh=H
load start
prbprb(PFA,PD,hh)
fig
```

SNRCALC

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called snrcalc designed to calculate the signal to noise
% voltage ratio of a target in random white Gaussian noise.
%
%
%  INPUTS
% sig    (1,1)       = Input pixel value of target
% noise  (1,1)       = Input pixel value of noise
% D      (1,1)       = Diameter of circular disc to be used in
%                      defining the filter; D must be odd.
% G      (256,256)   = Background scene (RWG)
% Nsub   (1,N)       = Size of background scene to display
% filt   (Nsub,Nsub)= unit pillbox filter
%
%  OUTPUTS
% sigma  (1,1)       = standard deviation of the of the pixel values
%                      when no target was present. The noise.
% signal (1,1)       = mean value of the pixels when the target was
%                      present minus the mean value of the pixels
%                      when no target was present
% snr    (1,1)       = Signal to noise ratio
```

```
% snrnom (1,1)      = nominal signal to noise ratio
%
%
%   [sigma,signal,snr,snrnom]=snrcalc(sig,noise,D,G,,Nsub,filt);

function  [sigma,signal,snr,snrnom]=snrcalc(sig,noise,D,G,Nsub,filt);
R=(D-1)/2;
RN=(Nsub-1)/2;
x=(-RN:RN).^2;
x=ones(size(x'))*x;
R2=x+x';
ii=find(R2<=R^2);
L=length(ii);
pd=256-Nsub;
qd=256-Nsub;
Sum1=[];
N=1000;
for b=1:N
   x= (RN+1)+pd*rand(1,1);
   y= (RN+1)+qd*rand(1,1);
   Z= G(x-RN:x+RN,y-RN:y+RN);
   Sum1=[Sum1; sum(sum(Z.*filt))];
end
Sums=[Sum1*noise Sum1*noise+sig*L];
sigma=std(Sums(:,1));
back=mean(Sums(:,1));
targ=mean(Sums(:,2));
signal=targ-back;
snr=(signal)/sigma;
sinom=L*sig;
nonom = sqrt(L*(noise^2));
snrnom=sinom/nonom;
```

SNRESTER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called snrester used to calculate a set of random error
% results for the estimated signal to noise ratio for PD, PFA
% operator(random) performance results. The PD, PFA random
% performance resultsare drawn from a set whose distribution is
% governed by some nominal snr and the number of trials. It is
% assumed that half the trials havetargets and half do not, and that
% the operator's effective threshold is at half the nominal snr.
%
%
%  INPUTS
% N      (1,1) = number of trials
% snrnom(1,1) = nominal signal to noise ratio
% M      (1,1) = number of monte carlo simulations to be run
%
%
%  OUTPUTS
% sigsnr (1,1)= standard deviation of random snr results.
% snrbar (1,1)= mean value of random snr results.
% ransnr (1,M)= set of random snr results.
% PDM    (1,1)= Monte Carlo Probability of detection.
% PFAM   (1,1)= Monte Carlo probability of false alarm.
%
%
% [sigsnr,snrbar,PD,PFA,ransnr]=snrester(N,snrnom,M)

function [sigsnr,snrbar,PDM,PFAM,ransnr]=snrester(N,snrnom,M)
if ~exist('M'); M=1e3; end
p=(1-erf(snrnom/(2*sqrt(2))))/2;
q=1-p;
hN=N/2;
R=rand(1,hN*M);
```

```
ii=R<q;
ii=reshape(ii,hN,M);
TT=sum(ii);
ii=R<q;
ii=reshape(ii,hN,M);
NN=sum(ii);
ii=find(TT==hN|TT==0);
TT(ii)=[];
NN(ii)=[];
ii=find(NN==hN|NN==0);
TT(ii)=[];
NN(ii)=[];
PD=TT/hN;
PFA=1-NN/hN;
t2sig=erfinv(1-2*PFA);
s2sig=-erfinv(1-2*PD)+t2sig;
ransnr=s2sig*sqrt(2);
snrbar=mean(ransnr);
sigsnr=std(ransnr);
PDM=mean(PD);
PFAM=1-PDM;
```

MAKEPILL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called makepill used to generate a unit pillbox of
% target size.
%
%
%  INPUTS
% Nsub   (1,1)      = Size of the portion returned of whitened signal
%                     disc (an odd number)
% D    (1,1)        = Circle diameter (an odd number)
%
%  OUTPUTS
%
% pill  (Nsub,Nsub)= Box containing signal disc at center
%
% pill=makepill(Nsub,D)

function pill=makepill(Nsub,D);
R=(D-1)/2;
if rem(Nsub,2)==1
   x=((-(Nsub-1)/2:(Nsub-1)/2)).^2;
   x=ones(size(x'))*x;
   R2=x+x';
   ii=find(R2<=R^2);
   pill=zeros(Nsub,Nsub);
   pill(ii)=ones(size(ii));
else rem(Nsub,2)==0;
   x=(-Nsub/2:(Nsub/2-1)).^2;
   x=ones(size(x'))*x;
   R2=x+x';
   ii=find(R2<=R^2);
   pill=zeros(Nsub,Nsub);
   pill(ii)=ones(size(ii));
end
```

SCALING1

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called scaling1 used to scale the display to show pixels
% having a value some specified number of standard deviations
% about the mean.
%
%
%  INPUTS
```

```
% zs1    (51,51) = image with target
% mm     (1,1)   = mean value of the target region
% SD     (1,1)   = pixel standard deviation in sub without target
% scale  (1,1)   = half the number of standard deviations in full
%                  display dynamic range
% R      (1,1)   = Target circle radius
%
%       OUTPUTS
% zs1l   (51,51) = scaled and limitted version of sub
%
%   zs1l=scaling1(sub,mm,SD,scale,R)

function  zs1l=scaling1(zs1,mm,SD,scale,R);
b=1/(2*scale*SD);
a=-b*(mm-scale*SD);
zs1l=a+b*zs1;
jjl=find(zs1l<0);zs1l(jjl)=zeros(size(jjl));
jju=find(zs1l>1);zs1l(jju)=ones(size(jju));
```

SCALING2

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called scaling2 used to scale the display to show pixels
% having a value between the minimum value of background scene pixels
% and a value near the mean value of the background scene pixels.
%
%  INPUTS
% zs2    (51,51) = image with target
% mm     (1,1)   = mean value of the target region
% SD     (1,1)   = pixel standard deviation in sub without target
% scale  (1,1)   = the number of standard deviations in full display
%                  dynamic range
% R      (1,1)   = Target circle radius
%
%       OUTPUTS
% zs2l   (51,51) = scaled and limitted version of zs2
% scale  (1,1)   = the number of standard deviations in full display
%  [zs2l,scale] = scaling2(sub,mm,SD,scale,R)


function [zs2l,scale]=scaling2(zs2,mm,SD,scale,R);
F=1;
while F==1;
   x=(-R:(R-1)).^2;
   x=ones(size(x'))*x;
   R2=x+x';
   ii=find(R2<=R^2);
   msub=zs2;
   msub(ii)=ones(size(msub(ii)))*min(msub(ii(:)));
   MN=min(msub(:));
   MEAN=mean(msub(:));
   scale1=((MEAN)-MN)/6;
   MN2=MN+(scale+1)*scale1;
   zs2l=zs2;
   uu=find(zs2>=MN2);
   zs2l(uu)=ones(size(zs2l(uu)))*MN2;
   if min(min(zs2l))~=max(max(zs2l));
      F=0;
   else scale=scale*1.5;
      end

   end
```

SCALING3

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called scaling3 used to scale the display to show pixels
```

```
% having a value between the maximum value of background scene pixels
% and the mean value of the background scene pixels.
%
%
%  INPUTS
% zs3    (51,51) = image with target
% mm     (1,1)   = mean value of the target region
% SD     (1,1)   = pixel standard deviation in sub without target
% scale  (1,1)   = half the number of standard deviations in full.
%                  display dynamic range
% R      (1,1)   = Target circle radius
%
%       OUTPUTS
% zs31   (51,51) = scaled and limitted version of sub
%
%  [zs31,scale]=scaling3(sub,mm,SD,scale,R)


function  [zs31,scale]=scaling3(zs3,mm,SD,scale,R);
F=1;
while F==1;
   x=(-R:(R-1)).^2;
   x=ones(size(x'))*x;
   R2=x+x';
   ii=find(R2<=R^2);
   msub=zs3;
   msub(ii)=zeros(size(msub(ii)));
   MX=max(msub(:));
   MEAN=mean(msub(:));
   scale1=(MX-(MEAN))/6;
   MX2=MX-((scale)*scale1);
   zs31=zs3;
   ll=find(zs3<=MEAN);
   zs31(ll)=ones(size(zs31(ll)))*MEAN;
   uu=find(zs3>=MX2);
   zs31(uu)=ones(size(zs31(uu)))*MX2;
   if min(min(zs31))~=max(max(zs31));
      F=0;
   else scale=(scale-.5);
      end

   end
```

SHOWR

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function called showr used to display in grayscale format
% a data array, with a specified position and size. used with real
% background noise. The program highlights the target area with a ring
% of target circumference.
%
%          INPUTS
%  subl    (N,N)  = data array for display
%  a       (1,1)  = x-axis position LL corner
%  b       (1,1)  = y-axis position LL corner
%  c       (1,1)  = x-axis image size
%  d       (1,1)  = y-axis image size
%  R       (1,1)  = circle radius
%
%
%      showr(subl,a,b,c,d,R)


function showr(subl,a,b,c,d,R)
x=(-R:(R-1)).^2;
x=ones(size(x'))*x;
```

```
R2=x+x';
ii=find(R2<=R^2);
AX = axes('position',[a b c d]);
axes(AX);
axis off;
colormap(bone);
caxis('auto');
pcolor(subl);
shading flat
hold on;
[M,N]=size(subl);
t=linspace(0,2*pi,100);cy=(R+1)*sin(t);cx=(R+1)*cos(t);
plot(cx+(N/2+1),cy+(M/2+1),'y:');
cx=ones(1,100); cy=linspace(1,N,100);
plot(cx,cy,'y');
cx=N*ones(1,100); cy=linspace(1,N,100);
plot(cx,cy,'y');
cy=ones(1,100); cx=linspace(1,N,100);
plot(cx,cy,'y');
cy=N*ones(1,100); cx=linspace(1,N,100);
plot(cx,cy,'y');
hold off;
axis off;

```

WRITEFIL

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% function called writefile designed to write experimental data
% to a .mat file and a .txt file for future use. The .mat files are
% called opwdata_.mat and the text files are called dataopw_.
%
%
%  INPUTS
%    n          (1,1)      = number of displays
%   Nsub        (1,1)      = size of the background scene for display.
%    D          (1,1)      = circle Diameter
%   sig         (1,1)      = Target signal
%  noise        (1,1)      = Target noise
%  snrnom       (1,1)      = nominal signal to noise voltage ratio
%  sigsnr       (1,1)      = standard deviation of the signal to noise
%                            ratio
%    c1         (2,1)      = number of times target was present and
%                            detected
%    c2         (2,1)      = number of times target was present and not
%                            detected
%    c3         (2,1)      = number of times no target and no detection
%    c4         (2,1)      = number of times no target and false detection
%    PDO        (1,1)      = Probability the target was present and the
%                            operator identified it
%    PFAO       (1,1)      = Probability the target was not present but
%                            the operator thought it was present
% snrexp        (1,1)      = Signal to noise voltage ratio of the operator
%                            during the experiment
%
%
%  OUTPUTS
% none
%
% writefil(n,c1,c2,c3,c4,snrexp,PDO,PFAO,sigsnr,sig,snrnom,D,Nsub)

function writefil(n,c1,c2,c3,c4,snrexp,PDO,PFAO,sigsnr,sig,snrnom,D,Nsub)
write=input('Is this a new file ? [y if a new; any key if not ] >>','s');
if write=='y';
   load fnum;
   fnum=fnum+1;
   save fnum fnum;
```

```
41     fid=fopen(sprintf('dataopw#%1.0f',fnum),'w');
42     operator=input('please enter your first and last name >>','s');
43     fprintf(fid,operator);
44  fprintf(sprintf('dataopw#%1.0f',fnum),'\nn=%1.0f\nc1=%1.0f ...
45  \nc2=%1.0f\nc3=%1.0f\nc4=%1.0f\nsnrexp=%7.6f\nPDO=%7.6f ...
46  \nPFAO=%7.6f\nsigsnr%7.6f\nsig=%4.3f\nsnrnom=%5.4f\nD=%1.0f ...
47  \nNsub=%1.0f\n\n',n,c1,c2,c3,c4,snrexp,PDO,PFAO,sigsnr,sig, ...
48  snrnom,D,Nsub);
49  else
50     load fnum
51  fprintf(sprintf('dataopw#%1.0f',fnum),'\nn=%1.0f\nc1=%1.0f ...
52     \nc2=%1.0f\nc3=%1.0f\nc4=%1.0f\nsnrexp=%7.6f\nPDO=%7.6f ...
53     \nPFAO=%7.6f\nsigsnr%5.6f\nsig=%1.0f\nsnrnom=%5.4f ...
54     \nD=%1.0f\nNsub=%1.0f\n\n',n,c1,c2,c3,c4,snrexp,PDO, ...
55     PFAO,sigsnr,sig,snrnom,D,Nsub);
56  end
57
58
59
```

CHECK

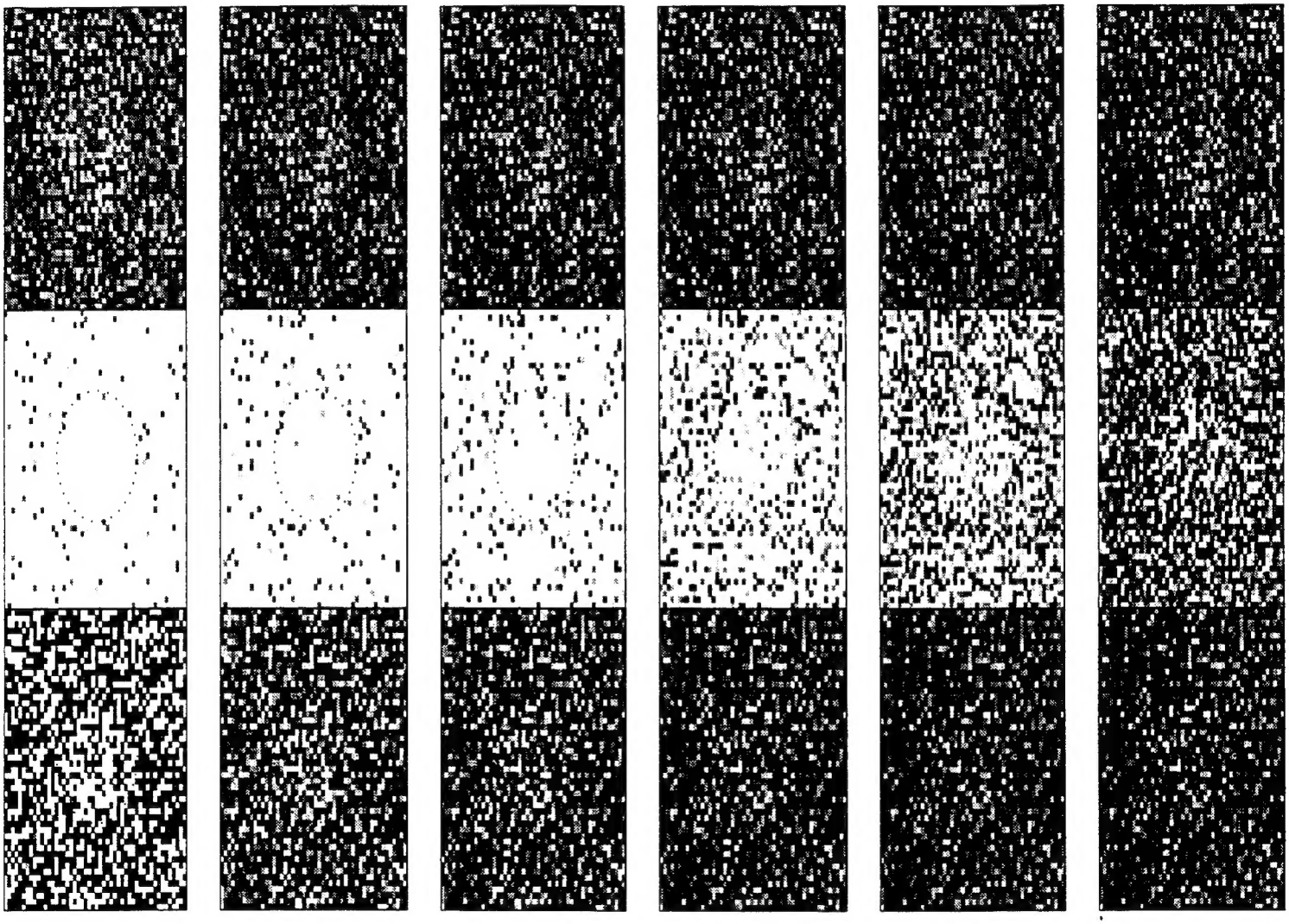
```
 1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2  %  function called check used to ensure that in an experiment an
 3  %  operator performs within an acceptable limit (two standard
 4  %  deviations) of the nominal signal to noise voltage ratio.
 5  %              INPUTS
 6  %    snrnom    (1,1)   = nominal signal to noise voltage ratio
 7  %     c1       (2,1)   = number of times target was present and
 8  %                        detected
 9  %     c2       (2,1)   = number of times target was present and not
10  %                        detected
11  %     c3       (2,1)   = number of times no target and no detection
12  %     c4       (2,1)   = number of times no target and false detection
13  %   sigsnr     (1,1)   = standard deviation of the signal to noise ratio
14  %
15  %
16  %        OUTPUTS
17  %     PDO      (1,1)   = Probability the target was present and the
18  %                        operator identified it
19  %     PFAO     (1,1)   = Probability the target was not present but
20  %                        the operator thought it was present
21  %   snrexp     (1,1)   = Signal to noise voltage ratio of the operator during
22  %                        the experiment
23  %   flag       (1,1)   = experimental exit flag. flag  =1 stops experiment
24  %
25  %    [PDO,PFAO,snrexp,flag]=check(c1,c2,c3,c4,snrnom,sigsnr)
26
27  function [PDO,PFAO,snrexp,flag]=check(c1,c2,c3,c4,snrnom,sigsnr);
28  PDO=c1/(c1+c2);
29  PFAO=c4/(c3+c4);
30  t2sig=erfinv(1-2*PFAO);
31  snrexp=sqrt(2)*(t2sig-erfinv(1-2*PDO));
32  lower=snrnom-1.96*sigsnr;
33  upper=snrnom+1.96*sigsnr;
34     if snrexp< lower | snrexp > upper;
35        disp('you are out of tolerance');
36        sprintf('your snrexp is : %5.4f',snrexp)
37        sprintf('the acceptable tolerance is : %5.4f to %5.4f',lower, ...
38                upper)
39        cont=input('do you want to stop?[y stops the test, any key to ...
40                  continue ] >>', 's');
41        if(cont=='y');
42           flag=1;
43        else
44           flag=0;
45        end
```

```
46      else sprintf('GOOD JOB! your snrexp is : %5.4f',snrexp)
47         flag=0;
48      end
49
```

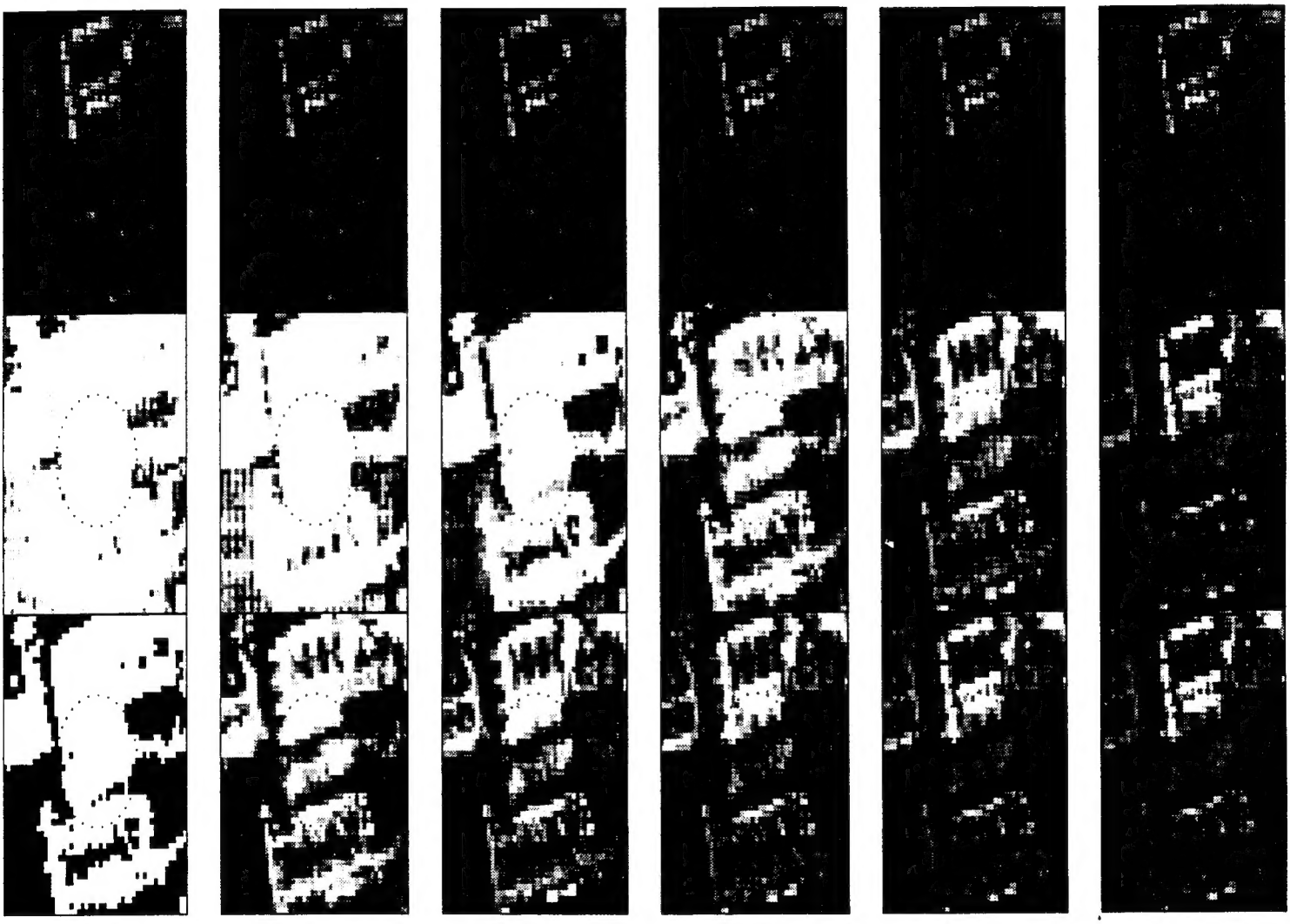
APPENDIX D. MONITOR DISPLAY

This appendix contains greyscale representations of the monitor display shown to the operator for each of the experiments In each display there is a target present however the reader may not percieve the target in every display. In the experiments the operator only had to identify a target in a single display. If he could see the target in any of the eighteen greyscale representationsthen he made the decision that the target was present.

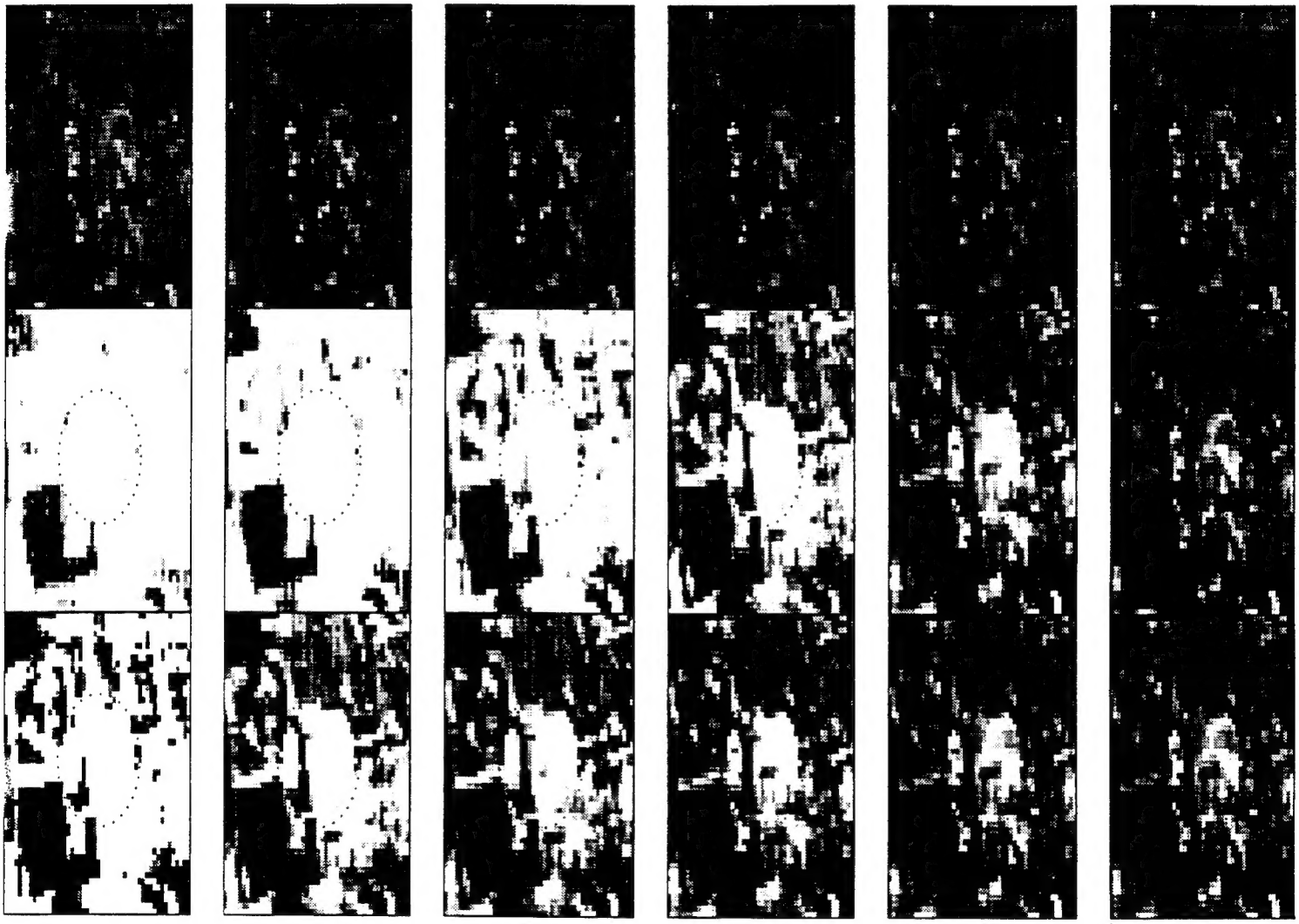
EXPERIMENT 1 MONITOR DISPLAY



EXPERIMENT 2 MONITOR DISPLAY



EXPERIMENT 3 MONITOR DISPLAY



INITIAL DISTRIBUTION LIST

1.Defense Technical Information Center 2
Cameron Station, Bldng 5
Alexandria, Virginia 22304-6145

2.Library, Code 52 2
Naval Postgraduate School,
Monterey, California 93943-5101

3.Prof Fried D. L. 1
Dept. Physics, Code PH/Fd
Naval Postgraduate School,
Monterey, California 93943-5002

4.Prof Davis D. S. 1
Dept. Physics, Code PH/Dv
Naval Postgraduate School,
Monterey, California 93943-5002

5.Prof Colson W. B. 1
Chairman Dept. Physics, Code PH/Cw
Naval Postgraduate School,
Monterey, California 93943-5002